

# Modeling in MathWorks Simscape by building a model of an automatic gearbox

---

Staffan Enocksson



UPPSALA  
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet  
UTH-enheten**

Besöksadress:  
Ängströmlaboratoriet  
Lägerhyddsvägen 1  
Hus 4, Plan 0

Postadress:  
Box 536  
751 21 Uppsala

Telefon:  
018 – 471 30 03

Telefax:  
018 – 471 30 00

Hemsida:  
<http://www.teknat.uu.se/student>

## Abstract

### **Modeling in MathWorks Simscape by building a model of an automatic gearbox**

*Staffan Enocksson*

The purpose of this thesis work has been to analyze the usability and the feasibility for modeling with MathWorks simulation tool Simscape by building a simplified model of the automatic gearbox ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C). It has been shown throughout the thesis how this model is built. First has system knowledge been acquired by studying relevant literature and speaking with the persons concerned. The second step was to get acquainted with Simscape and the physical network approach. The physical network approach that is accessible through the Simscape language makes it easy to build custom made components with means of physical and mathematical relationships. With this background a stepwise approach has been conducted which has led to the final model of the gearbox and the validation concept.

The results from this thesis work indicate that Simscape is a powerful tool for modeling physical systems and the results of the model validation give a good sign that it is possible to build and simulate physical models with the Simscape software. However, during the modeling of the ZF-ECOMAT 4 some things have been discovered which could improve the usability of the tool and make the learning curve for an inexperienced user of physical modeling tools less steep. In particular, a larger model library should be included from the beginning, more examples of simple and more complex models, the object-oriented related parts such as own MATLAB functions should be expanded, and a better troubleshooting guidance.

Handledare: Afram Kourie  
Ämnesgranskare: Bengt Carlsson  
Examinator: Elisabet Andréddóttir  
ISSN: 1650-8319, UPTec STS11 017

## Populärvetenskaplig beskrivning

Syftet med den här uppsatsen har varit att analysera användbarheten och möjligheten att modellera med MathWorks simuleringsverktyg Simscape genom att bygga en förenklad modell av den automatiska växellådan ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C). Genom uppsatsen har det visats hur denna modell är uppbyggd. Först har en systemkunskap inhämtats genom att studera relevant litteratur och genom att tala med berörda personer. Det andra steget var att bekanta sig med Simscape och den fysiska modelleringsapproachen. Den fysiska modelleringsapproachen som är tillgänglig via Simscape-språket gör det enkelt att bygga egentillverkade komponenter med hjälp av fysiska och matematiska samband. Med den här bakgrunden har en stegvis tillvägagångssätt genomförts vilket har mynnat ut i den slutgiltiga modellen av växellådan och valideringskonceptet.

Simscape har visat sig vara ett kraftfullt verktyg för att modellera fysikaliska system och resultatet från modellvalideringen ger en god indikation att det är möjligt att bygga och simulera fysikaliska modeller med Simscape-mjukvaran. Dock ska det nämnas, att under modelleringen av ZF-ECOMAT 4 så dök det upp saker som skulle kunna öka användbarheten av verktyget och minska inlärningskurvan för en ovan användare av fysikaliska modelleringsverktyg. Framförallt att ett större modellbibliotek borde finnas med från början, mer exempel av enkla och mer komplicerade modeller, de objektorienterade delarna som t.ex. egna MATLAB-funktioner borde byggas ut, samt en bättre felsökningsguide.

## **Acknowledgements**

This master thesis has been carried out with great satisfaction at the RBNP department at Scania's Research and Development facility between September 2010 and February 2011. It is the final piece in my engineering degree in Sociotechnical Systems Engineering (STS) at Uppsala University.

First and foremost I would like to give a huge thank to my supervisor at RBNP, Afram Kourie who has given me a great support and guidance throughout the whole thesis. He always made sure we were on the right track and corrected every small error.

Secondly, I would like to thank two persons who gave a good kick start with the thesis; Patrik Ekvall at MathWorks who introduced me into the world of physical modeling and Niklas Berglund at RBNP who patiently described the ZF-ECOMAT 4 and its components.

Thirdly, I would give a huge thank to all of the people at RBNP for a pleasant and very educational time.

And at last I would like to thank both my examiner Elisabet Andresdottir and subject reviewer Bengt Carlsson at Uppsala University.

Staffan Enocksson

Södertälje 11-02-23

## Table of Contents

1	Introduction.....	3
1.1	Purpose.....	4
1.2	Goals.....	4
1.3	Delimitations .....	4
2	Method.....	5
2.1	The modeling phase .....	5
2.2	Modeling and simulation tools used .....	6
2.3	Time requirements .....	7
3	Transmissions in general .....	8
3.1	The function of the transmission .....	8
3.2	ZF-ECOMAT Transmission.....	8
3.2.1	The Clutch.....	10
3.2.2	The Torque Converter .....	10
3.2.3	The Retarder .....	11
3.2.4	The Planetary Gear Sets .....	12
4	Modeling and simulation.....	13
4.1	Different kinds of modeling approaches.....	15
4.2	Model verifying.....	15
4.3	Which requirements should be considered for a modeling tool? .....	16
5	Modeling in Simscape.....	17
5.1	Across variable .....	17
5.2	Trough variable.....	17
5.3	Direction of variables .....	18
5.4	Connector ports and Connection Lines .....	19
5.5	Simscape language .....	19
5.6	The Simscape Library.....	22
5.7	Compilation and troubleshooting .....	23
6	Modeling of the ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C) .....	25
6.1	The Clutch model.....	25
6.2	The Torque Converter model .....	28
6.2.1	The Lockup-clutch.....	30
6.3	The Retarder model.....	30

6.4	The Planetary Gear Train model.....	33
6.4.1	Planetary gear sets .....	34
6.4.2	The full gear model.....	35
6.4.3	Gear Ratios .....	36
6.5	The Automatic Logic program .....	37
7	Validation .....	39
7.1	Remaining approximations.....	39
8	Results .....	42
8.1	Test 1, with the automatic shift logic: .....	42
8.2	Test 2, with the recorded gear shift signal:.....	44
8.3	Results discussion .....	45
9	Conclusion .....	46
9.1	Recommendations for future work.....	49
10	Bibliography.....	50
11	Appendix.....	52
11.1	Simscape MATLAB Supported Functions.....	52
11.2	Complete Simscape code .....	53
11.2.1	The Ideal Gear model .....	53
11.2.2	The Clutch model.....	53
11.2.3	The Torque Converter model .....	54
11.2.4	The Planetary Gear model.....	55
11.3	Parameters setting used.....	56
11.3.1	Clutches .....	56
11.3.2	Torque Converter .....	56
11.3.3	Retarder.....	56
11.3.4	Planetary gears.....	56
11.3.5	Final gear .....	56
11.3.6	Automatic Logic Program .....	57
11.3.7	Air drag .....	57
11.3.8	Rolling resistance.....	57
11.3.9	Other parameters.....	57
11.4	Torque output on propeller shaft .....	58
11.5	The final validation concept .....	59

# **1 Introduction**

The requirements for developing and testing new products have never been higher, especially for many manufacturing industries. Customers, competitors and regulatory boards are setting standards for new products that are going to be used in the society for a variety of different purposes. One industry where the requirements have escalated in a number of fields in the recent years is the automotive industry.

Particularly it is the transport sector that has been affected with increasing requirements for alternative fuels, decreased emission levels and engine efficiency. More and more goods and people are to be transported each day in increasingly shorter times. The automotive industry is trying each day to cope with these demands. (European Automobile Industry Report, 2009-2010)

With long and costly developing processes combined with the increasing demands and at the same time as computers and software have gotten faster has led to more investments in the field of modeling and simulation. (Engineering Simulation Solutions for the automotive Industry, 2008)

Simulation used to be performed entirely by experts in the field using expensive and dedicated computer systems. Today significant simulations can be performed on personal computers by experts in a specific field without the need for a staff of simulation specialists. Modern languages, tools and architectures have become better, more specialized and more user friendly. Many of these tools can today encapsulate much of the traditionally difficult work in building models and the main necessity today for building complex models of reality is mainly knowledge about the system in focus. (SMITH, Roger D., 2003)

The automotive industry has followed down the same path with huge investments in new technology. Going from an industry, consisting of more or less only mechanics to progress into an industry where computer technology is involved every day, both in the trucks and in the daily work. (ZACKRISSON, Tomas, 2003)

Computer simulations are, as mentioned above, one part which has increased rapidly in a lot of different fields in the automotive industry. It has become extremely important to test components in simulations to find possible design errors before building real prototypes. In many cases it has proven to be more cost effective, shorter development processes, less dangerous, or otherwise more practical than testing the real system. In the end this will hopefully lead to products with higher quality, shorter time to market processes and meet the required standards. (SMITH, Roger D., 2003)

Most of the vehicles being developed today at Scania CV AB in Södertälje consist of a series of different systems and components which has become increasingly advanced. This modular system makes it possible for Scania to produce different kind of vehicles optimized for a specific user need and at the same time as costs can be kept at a low level for development, production and spare parts management. (Scania.se)

In the continuing development process Scania has progressed with their modular thinking by building a model library of different vehicle components which goes by the acronym STARS. The acronym stands for Scania Truck and Road Simulation and consists of a simulation tool with a graphical user interface and compiled models of complete vehicles. The library consists of models of vehicle components such as combustion engines, gearboxes, axles, wheels, tires etc. STARS is used to make good estimates of fuel consumption, emissions and shorten lead periods for different driving scenarios and distances. The models are like the truck and buses also built in modules so they can be developed separately and then put together into a complete working vehicle models.

The library is in the process of constant development and in the further development process of components that go into production every single day there are new demands set for the simulation tools in translating this components into effective models. To be able to build complex models of different vehicle components efficiently, high demands are therefore set on the usability of the new simulation tools.

## **1.1 Purpose**

The purpose of this master thesis is to analyze the usability and the feasibility for modeling with MathWorks simulation tool Simscape by building a simplified model of the automatic gearbox ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C).

## **1.2 Goals**

- To get an understanding of how to model with Simscape simulation software.
- To model an automatic gearbox in the Simscape environment by means of physical and mathematical relationships and technical data.
- General research about Simscape's modeling potential in respect to usability, compilation/troubleshooting and simulation ability.

## **1.3 Delimitations**

Due to the purpose of this thesis all components models are kept simple, which implies:

- No static friction is accounted for in the clutch model
- No fluid drag losses is accounted for in the torque converter mode
- No bearing or mesh losses are accounted for in the planetary gears
- No hydraulics will be modeled
- No elastic driveline will be used
- The final complete vehicle model is only validated against a reference vehicle. No separate components have gone through any validation process, except analytically.

## 2 Method

This master thesis has been conducted at the RBNP department at Scania's R&D facility. RBNP has the main responsibility for the drivability of the powertrain for buses. Much of the daily work consists of simulating and test driving of the buses from a performance perspective with the help of tools such as simulation models and measuring computers.

The first step in the master thesis was acquiring knowledge about the gearbox system. Interviews with Niklas Berglund<sup>1</sup> were made to be able to understand what an automatic gearbox is and the function of its components.

The second step was to get a theoretical perspective. By doing a desktop research with specific search keywords like transmission, gearbox, planetary gear, simulation and modeling a broad field of different literature could be gathered. Thereafter a literature review was made of the collected material to get a deeper understanding of the specific components that was going to be included in the system and about the modeling and simulation concept. Both the Internet, books, drawings, technical documents etc. was used as source of information.

The third step was to get acquainted with the simulation tool Simscape. By reading the instruction manuals from MathWorks homepage ( (Simscape™ 3 User's Guide, 2010), (Simscape™ 3 Language Guide, 2010) ) and by looking at recorded webinars posted by MathWorks an initial shallow understanding of the physical network modeling approach could be reached.

During the fourth week a workshop was held at Scania. Patrik Ekvall (a Mathworks representative), came and talked about the features of Simscape and how it could be of use in the modeling part. Three web-meetings were thereafter scheduled. During the web-meetings we discussed the problems that I had encountered, whether they were principle or simulation tool specific. Especially he taught me how to think when you are dealing with physical modeling and he also helped me with the modeling of the clutch.

Throughout the thesis writing continuous meetings at random time interval has also been made with my supervisor at Scania, Afram Kourie. He has worked as a sound board for me to discuss new ideas and problems that have arisen.

### 2.1 The modeling phase

The modeling phase has been about understanding and trying to model the systems behavior analytically. It has also been carried out incrementally with a lot of trial and error. Each component has therefore been tested separately to verify it worked the way it was expected to do analytically, before moving on to the next component. Each component has also been tested together with one another, starting with two components, and then adding one after another. The process has been iterative in which both forward and backward steps have been taken. This incremental stepwise time consuming approach made the troubleshooting process a whole lot easier when it was time to simulate the whole model configuration.

---

<sup>1</sup> Niklas Berglund working at RBNP has a background at Scania with manual and automatic transmission, both in production and implementation/calibration in bus chassis.

## 2.2 Modeling and simulation tools used

All tools that are going to be used in this thesis are developed by MathWorks®<sup>2</sup>. Below is a brief description of the four tools used in this thesis.

### **MATLAB®**

Version 7.9.0.529 (R2009b) 12-Aug-2009

MATLAB (matrix laboratory) is a well-known numerical computing environment and a fourth-generation programming language developed by MathWorks. MATLAB is used for a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. MATLAB is very common among engineers and is taught at universities all over the world.

### **Simulink®**

Version 7.4 (R2009b) 29-Jun-2009

Simulink is a commercial tool for modeling, simulating and analyzing multi-domain dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries. Simulink and MATLAB are tightly integrated and Simulink can either drive MATLAB or be scripted from it. It is regularly used for designing, simulating, implementing and testing of variety of time-varying systems such as communication, control theory, digital signal processing etc.

### **Simscape™**

Version 3.2 (R2009b) 29-Jun-2009

Simscape offers a MATLAB-based, object-oriented, physical modeling language for use in the Simulink environment. Simscape is a software extension for MathWorks Simulink and provides tools for modeling systems spanning mechanical, electrical, hydraulic, and other physical domains as physical networks. From these different physical domains you can create models of your own custom components. Simscape provides a set of block libraries and special simulation features especially for modeling physical systems that consists of real physical components. It is accessible as a library within the Simulink environment.

### **Stateflow®**

Version 7.4 (R2009b) 29-Jun-2009

Stateflow is a design environment for developing state charts and flow diagrams. It provides elements for describing complex logic in a natural, readable and in an intuitive form. It is also tightly integrated with MATLAB, Simulink and Simscape.

---

<sup>2</sup> MATLAB, Simulink, Stateflow are registered trademarks, and Simscape is a trademark of The MathWorks, Inc.

## 2.3 Time requirements

The time frame requirements for this thesis work are presented in Table 1, where each step and its corresponding time requirement are listed.

**Table 1: Time requirements table**

<b>Week</b>	<b>Activity</b>
1-2	Introduction Starting with report
3-4	Literature review Writing report
4-8	Starting to get acquainted with Simscape Starting to build the components
8-15	Building the model Writing report
15-18	Finishing the model Validation of model Writing report
18-19	Finishing report
19-20	Finishing report Making presentation

### 3 Transmissions in general

In this section the automatic gearbox and its components are described.

#### 3.1 The function of the transmission

The basic function of the ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C) transmission, or any other transmission or gearbox, is to enable angular motion and torque conversion from a rotating power source (combustion engine, electrical engine etc.) to another device (wheel, shaft etc.) using different kinds of gear configurations. The purpose of the gearbox is to convert the engines rotating momentum to an appropriate angular speed and torque to the driving wheels. Combustion engines needs in most cases to operate at a relatively high rotational speed, which does not work very well for starting, stopping and slower travel. The transmission converts the higher engine speed (rpm) to the slower wheel speed with an increase in torque which gives the vehicle a different driving range, from hill climbs, to crawling and for going 100 km/h on a freeway.

Practically a gear configuration works like this:

A small gear against a big decreases the rpm value but increases the torque power and vice versa, a big gear against a small increases the rpm value but with less torque power. Usually there is also a reverse gear which shifts the direction of the rotation of the driving wheels in the opposite direction. (BOSCH, 2000; Wikipedia--Transmission (mechanics))

Multi-speed gearboxes have become the established standard of power transmission in many modern motor vehicles today. Shifting on multi-speed gearboxes is performed using either disengagement of power transmission (manual and semi-automatic transmission) or under load by a friction mechanism (automatic transmissions). Common for automatic transmissions is that the driver doesn't have to worry about shifting gears. A frequent application for the automatic transmission with friction mechanism is when there is a lot of stop and go traffic which require a lot of frequent gear shifts without excessive comfort disorder. They are especially used for many city vehicles. (BOSCH, 2007; Wikipedia--Automatic\_transmission)

#### 3.2 ZF-ECOMAT Transmission

ZF Friedrichshafen AG (ZF) manufactures and produces among other transmissions an automatic transmission series called ECOMAT. The gearboxes in the ECOMAT series are currently in third generation which goes by the name ECOMAT 4. In the table below are the existing three different models listed in the ECOMAT 4 series.

3rd generation — ECOMAT 4 (2006-present)

- 5HP-504 / 6HP-504 — five- or six-speed; maximum input torque of 1,100 newton metres (811 ft·lbf)
- 5HP-594 / 6HP-594 — five- or six-speed; maximum input torque of 1,250 newton metres (922 ft·lbf)
- 5HP-604 / 6HP-604 — five- or six-speed; maximum input torque of 1,750 newton metres (1,291 ft·lbf)

The gearboxes are used in many commercial and special vehicle applications and can be designed with the choice of 5 and 6-speed versions. Possible applications for the gearboxes are everything from city buses to coaches. In Figure 1 the actual ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C) is shown. Most modern automatic gearboxes have almost the same components but the configurations can differ (depending on type). A detailed explanation of the components that will be included in the modeling chapter will thereafter be presented. (ZF FRIEDRICHSHAFEN AG , 2006; Wikipedia--List\_of\_ZF\_transmissions; BOSCH, 2007)

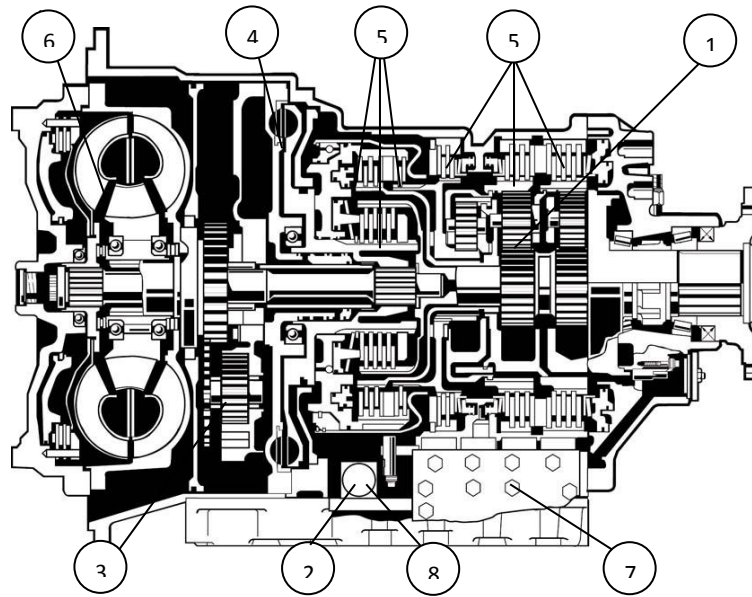


Figure 1: ZF-ECOMAT (HP 504 C / HP 594 C / HP 604 C)

Table 2: Automatic transmissions components description

Number	Component	Function
1	Planetary Gear Sets	Sets the various conversion ratios
2	Hydraulic system	Complex maze of passage and tubes that sends transmission fluid under pressure to all parts of the transmission.
3	Oil Pump	Engine driven pump that pressurizes the hydraulic fluid. It also supports the lubrication and cooling system in the transmission.
4	Retarder	A non- wearing brake
5	Clutches and brakes (the difference is that if the driven member is fixed to its frame, it is called a brake)	Effect gear changes without interrupting the flow of power.
6	Torque Converter (with lock-up clutch)	Transfer speed and torque and keeps the engine from stopping at low speeds
7	Transmission shift control unit	Defines the gear selections and shift points.
8	Hydraulic and lubricating oil	Provides lubrication which prevents corrosion

### 3.2.1 The Clutch

The first component that will be introduced is the clutch. A clutch is a mechanical device that provides a smooth and gradual connection between two separate members rotating at different speed about a common axis. Most of them consist of a number of friction discs which are pressed tightly together in a clutch drum. There are two types of frictions clutches (dry-plate and wet-plate), where wet-plate friction clutches have better thermal performance but worse drag losses. The clutch can connect the two shafts so that they either can be locked together and spin at the same speed or be decoupled and spin at different speeds.

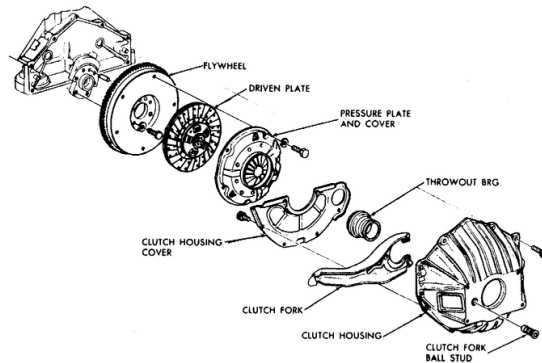


Figure 2: Exploded view of a typical clutch

A clutch works in the following way (see Figure 2 above for details); a pressure source applies the force which joins the flywheel, pressure plate and driven plate for common rotation (engaged mode). The clutch is disengaged by a mechanical or hydraulic actuated throw-out bearing applies force to the center of the pressure plates, thereby releasing the pressure at the periphery. Clutches engagement/disengagement are either controlled by a clutch pedal or by an automatic control unit. A torsion damper/coupling may be integrated to reduce vibrations in the driveline. (BOSCH, 2007; Wikipedia--Clutch)

### 3.2.2 The Torque Converter

The function of the torque converter is like the gearbox to transfer rotating power to another driven load efficiently and at the same time smoothly. It also allows the engine to keep on rotating at an idle speed when the vehicle comes to a stop without clutch operations. It consists of a fluid coupling which increases the lifespan of the gearbox since it decreases the frictional loss by converting it to heat. The fluid is often some kind of oil. There are three rotating elements: the impeller, the turbine wheel and the reaction element (stator). Torque is transferred from the engine's flywheel disk to the converter via a link which consists of flex plates or a torsion damper/coupling (standard in all Scania produced busses). (BERGLUND, Niklas, 2010)

A torque converter works in the following way (see Figure 3 for illustration); the impeller moves oil into a circular flow system which is controlled by the blades in the converter. The oil flow from the impeller side collides with the turbine wheel and is then diverted in the direction of flow. The purpose of the stator is to divert the oil flowing out of the turbine and providing it onwards to the impeller using suitable direction of flow. The stator experiences torque from the diversion and use this to increase the turbine rotating movement. The torque multiplying effect depends especially on the design of the blades in the converter and the viscosity of the liquid. (ZF FRIEDRICHSHAFEN AG , 2006; Wikipedia--Torque\_converter)

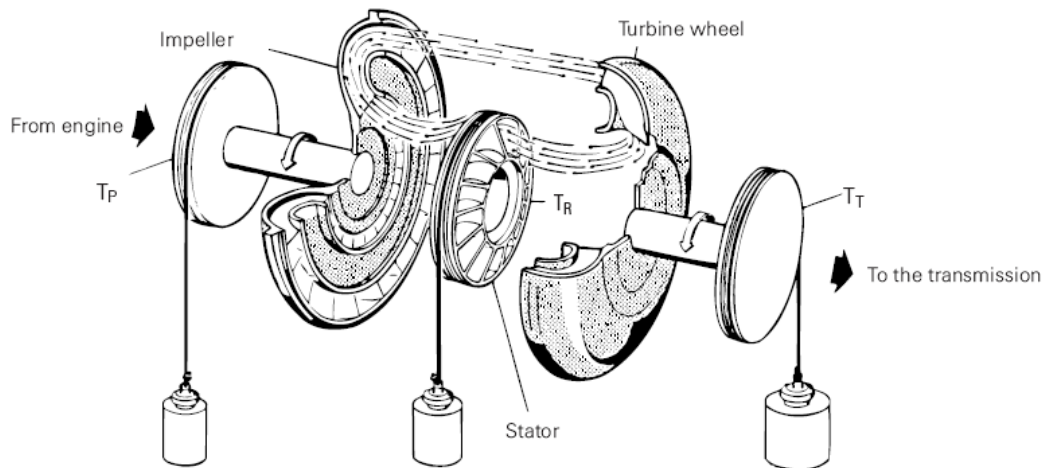


Figure 3: Torque Converter, Impeller, Stator and Turbine

### 3.2.3 The Retarder

A retarder is a non-wearing auxiliary brake that augments or replaces some of the functions of the primary braking system. It resembles a reversed torque converter in the way that it works as a fluid coupling and consists of a rotor and stator which forms a torus (see Figure 4). As in the torque converter, the fluid in the retarder is thrown between the rotor and stator which results in the braking torque. The retarder reduces the thermal load on the road wheel brakes under continuous braking, which is ideal for trucks or busses during descent of a long decline where the speed needs to be controlled and prolongs the life of the ordinary system. The retarder can either be hydrodynamic or electrodynamic and can be fitted on both the drive input side (primary retarders) or the output side (secondary retarders). Primary retarders can be mounted as an integrated unit in the transmission which allows for compact dimensions, low weight and fluid shared with the transmission in a single circuit. Integrated retarders are widely used on public transport buses because they have the above named specific design advantages and they are good for braking at low speeds, whilst secondary one's are often used in long-distance trucks for adjustment braking at higher speeds or when travelling downhill. (BOSCH, 2007)

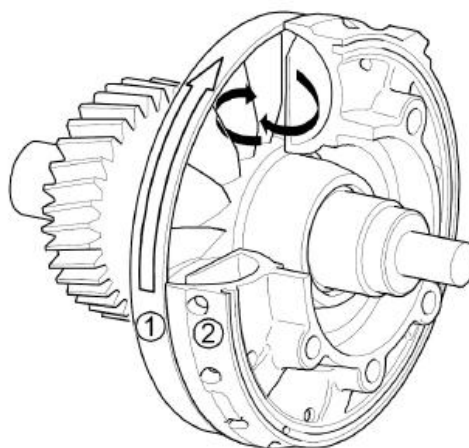


Figure 4: Retarder, 1 = Rotor, 2 = Stator

### 3.2.4 The Planetary Gear Sets

Planetary gear sets is a gear system that consists of planet gears revolving about a sun gear and an internal ring gear (see Figure 5 for illustration). It is characterized by at least one of the cog wheels in the gear system is mounted to an axis which is not fixed. The cog wheel can still move in a circle around the other cog wheel's fixed center. The planet gears are generally mounted on a carrier which can rotate relative to the sun gear. Each element can act as input or output gear, or it may be held stationary. That is why there are several ways in which an input rotation can be converted to an output rotation.

The layout of the planetary gear makes it ideal for use with friction clutches and brake bands, which are used for selective engagement or fixing of the individual elements in the planetary gear. The engagement pattern can be altered which change the conversion ratio without interrupting torque flow. In order to provide more conversion ratios (more gears) many planetary gear sets can be mounted in series, one after another in different arrangements. (BOSCH, 2007; Wikipedia--Automatic\_transmission)

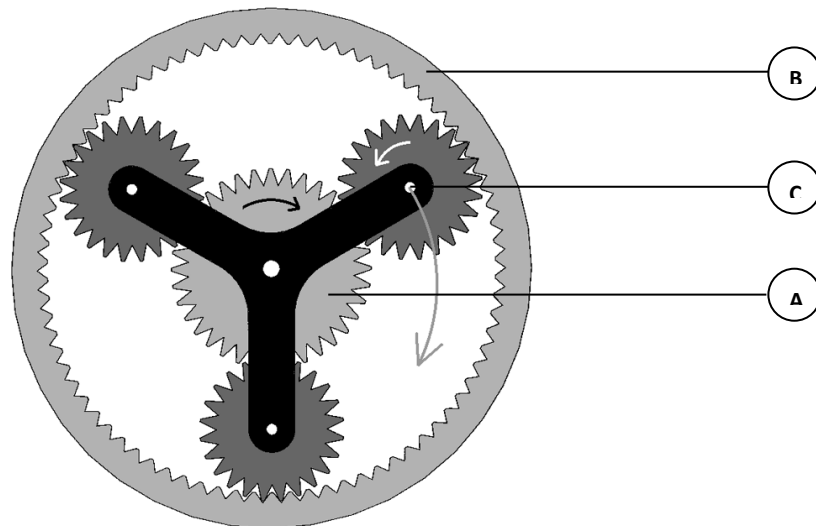


Figure 5: Planetary gear, A = Sun gear, B = Ring gear and C = Planet gears with carrier

## 4 Modeling and simulation

In the book *Encyclopedia of Computer Science*, Roger D. Smith<sup>3</sup> defines simulations as the process of designing a model of a real or imagined system and conducting experiments with that model. The purpose is to understand the behavior of the system and to link observations to understandable patterns. Since models build on representations of real world, assumptions are being made and mathematical algorithms and relationships are derived to describe these assumptions. Simulation is the imitation of reality and it builds on representations of certain key characteristics or behaviors of a selected physical system. Basically this is what all science is about, to describe the world around us. In many studies it has proven to be more cost effective, less dangerous, faster, or otherwisemore more practical than to test the real system. The system may not even (yet) exist. (SMITH, Roger D., 2003)

Dynamic processes are what characterize many systems in the real world. To be able to better understand and control them models are required. Models in these contexts are often built-up of mathematical equations, physical relations. In the end this can lead to good representations of the dynamic processes in the real world, either if they are simple linear relationships or non-linear. (LJUNG, Lennart and Glad, Torkel, 2008)

Ljung and Glad present two basic principles for how a model is constructed: physical modeling and system identification.

The first principle is to reestablish the real world properties and behaviours on subsystems. Different known laws of nature are used to describe the subsystems. What happens when you connect a gear to a rotational shaft is followed by Newtons laws about motion. If a system is simple the model may be represented and solved analytically with help of mathematical tools.

Consider the illustration in Figure 6 of a simple system of rotation of a rigid shaft connected to a driven member where Newtons laws are used to describe the physical system.



Figure 6: Rotation of a rigid shaft connected to a driven member

Trough Newtons second law of motion the following differntial equations is derived which decribe the rotation:

$$J\dot{\omega}(t) = \tau_d(t)$$

---

<sup>3</sup> Roger D. Smith is currently the CTO for Florida Hospital's Nicholson Center for Surgical Advancement, and has just finished as the CTO for US Army Simulation, Training and Instrumentation. He has published over 100 papers on innovation, management, technology and simulation (Amazon.com).

The expression can be reformulated if it is considered that  $\dot{\theta}(t) = \omega(t)$  to:

$$J\ddot{\theta}(t) = \tau_d(t)$$

where:

$\tau_d(t)$	<i>Torque driver</i>	[Nm]
$\theta(t)$	<i>Angle of the rotation</i>	[rad]
$\omega(t)$	<i>Angle of the rotation</i>	[rad/s]
$J$	<i>Moment of inertia (load)</i>	[kg * m <sup>2</sup> ]

However, problems in the real world are often more complex and many problems of interest can be so complex that it is impossible to make a simple analytical model representation. Coping with the complexity of the real world is a big challenge in modeling and simulation. For more complex systems such as a human being or the global climate; hypotheses and generally accepted relations are therefore used such as linear approximation. (SMITH, Roger D., 2003; LJUNG, Lennart and Glad, Torkel, 2008)

Another way to cope with the complexity of the real world is the other modeling principle; system identification or empirical modeling. The principle is based on observations of the system's behaviour in order to adjust the model's properties and behaviour to the system's. This principle is often used as a complement to the first one. Technical systems are initially built upon laws of nature that come from observations of subsystems. (LJUNG, Lennart and Glad, Torkel, 2008)

There is a distinction between two kinds of simulations, either discrete event or continuous, based on how the state variables change. Discrete events refer to that the state variables change at specific points in time and in a continuous simulation the state variables change continuously. Normally in a continuous simulation the variables are expressed in a function where time is one dimension of them. Most simulations use a combination of both discrete and continuous state variables, usually one of them is predominant and stands therefore for the classification of the whole system. (SMITH, Roger D., 2003)

## 4.1 Different kinds of modeling approaches

The traditional modeling methods (C, Fortran, etc.) and signal-based or input-output (Simulink) are often referred to as casual modeling tools. These tools work very well for control systems, but when it comes to physical systems they have some disadvantages. Physical systems are often expressed in the form of differential algebraic equations (DAEs), which are a composed set of equations, consisting of both derivatives and without, that must be solved simultaneously. Casual modeling tools can only approximate them and the models that are created are often dependent upon which element they are connected to. Therefore it is necessary to know which inputs and outputs that are available in order to connect it with the rest of the system. This leads to that every component has to be modelled in the same manner in order to reuse them in other systems or applications, especially when components span over multiple physical domains.

Because the above named reasons a new type of simulation tools grew based on acausal object-oriented physical modeling often referred to as non-casual or acausal modeling. Kirchhoff's laws had long been used to express the equations for an entire system of connected electrical components. Developers found that similar rules could be applied to other physical domains and with this came the rise of languages such as Simscape, Modelica, MapleSim and 20Sim. The advantages of these tools are particularly that the mathematical model does not depend upon location in the system making it easier to reuse component models, the equations for the network are created automatically which makes it easier to handle algebraic constraints and the non-casual approach makes modeling in multiple domains easier. A description of how Simscape apply this approach is followed in *Chapter 6 Modeling in Simscape*. (MILLER, Steve, 2008; Mathworks.com--Recorded Webinar: Physical Modeling with the Simscape Language)

## 4.2 Model verifying

It is trivial to build accurate models of representations of the real world, the difficulty lies in to build models with sufficient accuracy in order to give them credibility. Therefore every model needs to be tested and verified in order to give them acceptance. Model verifications is done by comparing the behaviour of the model against the real system and evaluate the difference. Ljung & Glad argues in their book *Modellbygge och simulering* that models have a certain areas of fidelity. Some models are valid for vague, qualitative, statements and others are valid for more precise, quantitative, predictions. The fidelity area responds to the model users accuracy requirements for the study. A model of a wind power station may for example only be valid for small breezes, but another one can be reliable for a hurricane wind. It is an impossibility to deal with every representation in a model, therefore a limit has to be set that is acceptable for the purpose of the study; what kind of variables to include/exclude. Maybe the model can be simplified by aggregating the effects of the excluded variables into the included ones. The bottom line here is that every model has certain level of fidelity, because every model are build on representations of the real world. Even though models and simulations are great for many reasons, it can never entirely replace observations and experiments. (SMITH, Roger D., 2003; LJUNG, Lennart and Glad, Torkel, 2008)

### **4.3 Which requirements should be considered for a modeling tool?**

In (LJUNG, Lennart and Glad, Torkel, 2008) a number of requirements that should be fulfilled for a modern modeling tool are listed:

- It should cover as many physical and technical domains as possible.
- It should be systematic. Ideally, large parts should be automated in the software.
- It should lead to a mathematical formulation that is appropriate for simulation and other modeling uses.
- It should be modular. It should therefore be possible to build component models that can then be assembled into complete systems.
- It should facilitate the reuse of models in the new context
- It should be close to the physics. It must therefore resemble the real physical world in an accessible way.

## 5 Modeling in Simscape

As mentioned earlier, Simscape is a non-casual or acausal modeling tool. Blocks in traditional modeling tools such as Simulink represent basic mathematical operators and when you connect blocks together you get a system of different mathematical operators with specific inputs and outputs. In Simscape each block in the system consists of functional elements that interact with each other by exchanging power or energy through their ports.

Connection ports in Simscape are bidirectional, where energy can flow in both directions. Connecting Simscape blocks represents connecting real physical components like shafts, valves etc. Flow direction and information flow does not have to be specified when connecting Simscape blocks into the network.

The number of connection ports for an element is determined by the number of energy flows it exchanges with other elements in the system. For example, a resistor can be characterized as a two-port element, with energy flow in and flow out. The resistor only involves one physical domain. Each energy flow is represented by its variables and each flow has two variables, one through and one across. In Simscape they are called basic or conjugate variables and for example in mechanical rotational systems there are torque and angular velocity. The difference between them is described below:

### 5.1 Across variable

Kirchhoff's voltage law states that the directed sum of the electrical potential differences around any closed circuit must be zero. This implies that the voltage of all components' ports attached to an electrical node must be the same. If this approach is transferred to the mechanical rotational domain it means that the angular velocity at all of the component's ports attached to that node must be the same.

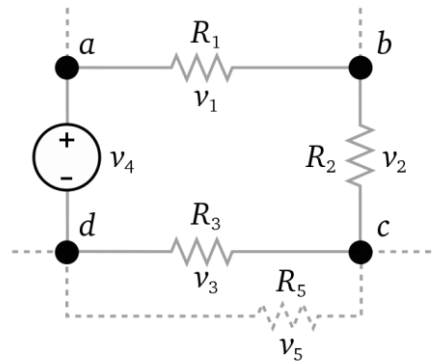


Figure 7: Across variable, the sum of all voltages around the loop is equal to zero.  $v_1 + v_2 + v_3 + v_4 + v_5 = 0$

### 5.2 Trough variable

Kirchhoff's current law states the sum of currents flowing towards an electrical node is equal to the sum of currents flowing away from the node. Once again if this approach is transferred to the mechanical rotational domain it means that the amount of torque flowing into that node must be equal to the amount flowing out.

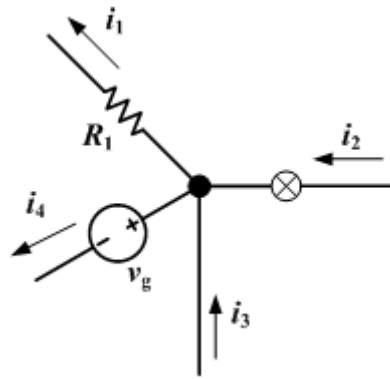


Figure 8: Trough variable, the current entering any junction is equal to the current leaving that junction.  $i_1 + i_4 = i_2 + i_3$

Expressing mathematical and physical equations for a component including these basic variables makes it possible to formulate the equations for the entire system in using this approach for each different physical domain. (MILLER, Steve, 2008)

In Table 3 are the predefined physical domains in the Simscape standard package listed with their respective trough and across variables. The variables are as described above are analogous to each other and the product of the variables are generally power (energy flow in watts), except for the pneumatic and magnetic domain where the product is energy. (Simscape™ 3 User's Guide, 2010)

Table 3: Simscape predefined physical domains

Physical Domain	Across Variable	Through Variable
Electrical	Voltage	Current
Hydraulic	Pressure	Flow rate
Magnetic	Magnetomotive force (mmf)	Flux
Mechanical rotational	Angular velocity	Torque
Mechanical translational	Translational velocity	Force
Pneumatic	Pressure and temperature	Mass flow rate and heat flow
Thermal	Temperature	Heat flow

### 5.3 Direction of variables

Every single variable in Simscape is represented with its magnitude and sign. In Figure 8 is an element with only two ports connected, and there is only one pair of variables, a trough and an across variable. The element is oriented from port A to port B meaning that the trough variable is positive if the flow is going from A to B. The across variable is defined as  $AV = AV_A - AV_B$ .

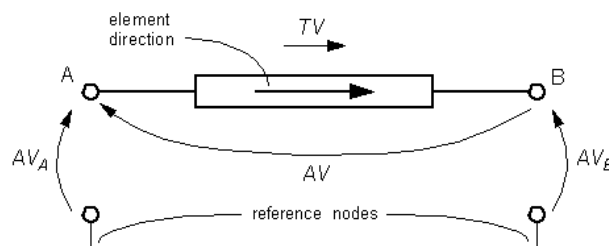


Figure 9: Simscape element, direction of variables

With this approach it is simple to determine the energy flow direction because the only thing that matters is the sign of the variables. It follows that their energy is positive if the element consumes energy and negative if it provides energy to the system. All network elements are separated into active and passive elements, depending on whether they deliver energy to the system, dissipates or store it. Therefore active elements as force and velocity sources and other actuators etc. must be oriented in line with the right action or function as they are expected to perform in the system. Passive elements like dampers, resistors, springs, pipelines etc. on the other can be oriented either way.

## 5.4 Connector ports and Connection Lines

Simscape has two different kinds of ports:



Physical conserving ports



Physical signal ports

Physical conserving ports are bidirectional, and the connections represent the physical connection with the exchange of energy flows. That is why only conserving ports can connect to other conserving ports of the same type and not to Simulink ports or Physical signal ports. Each different type of ports represents a physical domain. The lines that connect conserving ports are bidirectional lines that carry physical variables (through and across) instead of signals. Branching of physical connection lines are possible and in doing so any through variable transferred along the physical connection line is divided among the elements connected. Elements directly connected to each other continue to share the same across variables.

Physical signal ports are one-way directional and transfers signals that use an internal Simscape engine for computations. Physical signals are used instead of Simulink input and output ports to increase computation speed and avoid issues related to algebraic loops. The physical signals can have units assigned and Simscape deals with the necessary unit conversion operations if needed. (Simscape™ 3 User's Guide, 2010)

## 5.5 Simscape language

As mentioned earlier in the description of the Simulation tools, Simscape also has an object-oriented programming language tied to it. The language enables the user to create new self-defined components as textual files with equations represented as acausal implicit differential algebraic equations (DAEs). Each component can be used with another component if they share the same physical domain and if none of the predefined ones fit it is possible to create new ones.

The following example of an ideal gear can illustrate how the Simscape language works:

An ideal gear is a component that has an energy flow in and one out. It can be described physically with the following two equations, one for the angular velocity and one for the torque conversion:

$$\omega_{in} = ratio * \omega_{out} \quad (1.1)$$

$$\tau_{in} = -\frac{\tau_{out}}{ratio} \quad (1.2)$$

This component can easily be built with the Simscape language. The implementation and the description of the code and its components follow below. For the complete coherent code see Appendix 12.2.1 The Ideal Gear model.

#### Simscape code 1: The component's name and description section

```
component Ideal_gear
% Ideal Gear Description
```

Initially the component's name is declared. If needed, a description can thereafter be followed.

#### Simscape code 2: The node section

```
nodes
I = foundation.mechanical.rotational.rotational; % I:left
O = foundation.mechanical.rotational.rotational; % O:right
end
```

The node section is where the declaration of the component takes place and in this case there are two nodes that are associated with the mechanical rotational domain, which is one of the predefined physical domains in the standard Simscape package.

#### Simscape code 3: The parameter section

```
parameters
ratio = { 1, '1' }; % Min Gear ratio
end
```

Above are the component's parameters declared with their associated units. The parameter section defines the parameters that can only be changed before the simulation starts. Useful models will have parameters that correspond to actual physical quantities, which usually can be found in technical documents or measured. In this case the parameter ratio is set to a default value of 1.

#### Simscape code 4: The variables section

```
variables
t_in = { 0, 'N*m' };
t_out = { 0, 'N*m' };
w = { 0, 'rad/s' };
end
```

Here are the component's variables declared with their associated unit.

#### Simscape code 5: The function section

```
function setup
through( t_in, I.t, [] );
through( t_out, O.t, [] );
across( w, I.w, O.w);

% Parameter range checking
if ratio <= 0
    pm_error( 'Ratio value must be greater than zero' );
end
end
```

In the function setup section the relationships between the components variables (across and through) and its nodes are defined. Parameter validation can (if needed) also be implemented. In this case it checks that the ratio is always set to  $> 0$ .

#### Simscape code 6: The equation section

```
equations
I.w == ratio * O.w;
t_out == -ratio * t_in;
end
```

The last equation section defines the mathematical relationship between the components through and across variables, parameters, input/outputs and corresponding time derivative. In this case the relationships are between:

torque ( $t_{in}, t_{out}$ ), angular velocity ( $\omega$ ) and the unit-less parameter ( $ratio$ )

The double equal sign ( $==$ ) stands for continuous non-casual equality between the left and right side, not assignment or not a Boolean operator as in other programming languages. The equations are evaluated continuously and simultaneously throughout the simulation process. The equations can be DAEs or ODEs or both and can consist of vectors/matrices. Conditional equations can be specified using if statements.

All equations in Simscape are evaluated in continuous time. The values such as variables, inputs, outputs and time are defined as piecewise continuous. Piecewise continuous indicates that values are continuous over compact time intervals but may change at certain instances. Other values which are not time varying are, parameters and constants. Global simulation time is accessible in the equation section with the time function, and therefore the time derivative of an operand is also available. (Simscape™ 3 Language Guide, 2010)

The Simscape language supports some basic MATLAB functions which can be used in the equation section for example  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$ , (See Appendix 12.1 Simscape MATLAB Supported Functions for the complete list).

## 5.6 The Simscape Library

There is a standard library shipped with Simscape that consists of some basic components for the different physical domains. For example in the rotational mechanic domain the following components are included:

**Table 4: Rotational components with included source code provided with Simscape**

<code>Ideal Rotational Motion Sensor</code>
<code>Ideal Torque Sensor</code>
<code>Ideal Angular Velocity Source</code>
<code>Ideal Torque Source</code>
<code>Ideal Gear</code>
<code>Inertia</code>
<code>Mechanical Rotational Reference</code>
<code>Rotational Damper</code>
<code>Rotational Friction</code>
<code>Rotational Hard Stop</code>
<code>Rotational Spring</code>

As described earlier it is possible to build your own custom made physical component with the Simscape language and this custom built component can thereafter be gathered in a custom library, together with other own made components, similar to the provided library above.

Because Simscape is object-oriented modeling language, source code inheritance among the component is a feature provided. It is useful for building similar components that share the same basic variables and parameters, but need different parameter settings.

If the components are going to be distributed to other departments, companies etc., there is an option to protect the source code of the component for example if it contains sensitive information. (Simscape™ 3 Language Guide, 2010)

## 5.7 Compilation and troubleshooting

Troubleshooting in the Simscape is handled in a number of ways. Below is just an example of those that came up commonly during this thesis writing and those who are described in the Simscape User's guide and Simscape Language Guide.

The compiler or the solver gives an error if:

- The units are not commensurate

The Simscape code 7 example below would result in this error message because  $I.w$  has the unit  $[rad/s]$ ,  $t_{out}$  has  $[Nm]$  and  $ratio$  has  $[No\ unit]$  assigned, i.e. they are not commensurate.

**Simscape code 7: Not commensurate equation**

```
equations
I.w == ratio * t_out;
O.w == -ratio * t_in;
end
```

- More variables than equations (or the other way around)

For example you declare two variables in the setup section of the code and then have three equations in equations section. When the numerical solver tries to solve the system it gives an error message because the system of equations becomes over-determined or underdetermined and that causes numerical errors for the equation solver.

When studying systems of linear equations, the equations can either be linearly dependent or independent and if  $M$  is the number of equations,  $N$  is the number of unknowns and  $D$  is the linearly dependent equations. Table 5 illustrates the cases that determines if the system is determined, overdetermined or underdetermined. (Wikipedia--Overdetermined\_system)

**Table 5: Different equation systems**

$M = N$	Determined
$M < N$	Underdetermined
$M > N$ , and three special cases:	Overdetermined
When $M > N$ but all $M$ are not linearly independent, but when the linearly dependent equations $D$ are removed $M - D > N$ . This case yields no solutions.	Special Case 1, overdetermined
When $M > N$ but all $M$ are not linearly independent, but when the linearly dependent equations $D$ are removed $M - D = N$ . This case yields a single solution	Special Case 2, determined
When $M > N$ but all $M$ are not linearly	Special Case 3, underdetermined

independent, but when the linearly dependent equations  $D$  are removed  $M - D < N$ . This case yields infinitely many solutions.

- Other numerical issues

The most common problems were often related to numerical issues such as zero crossings which can be caused by certain configurations of Simscape blocks. Zero crossing is a specific event type represented by the value of a mathematical function changing sign (e.g. from positive to negative). Figure 10 illustrates how such an error may show up in Simscape, note that the error message does not say in which specific block the numerical error is caused or what the cause of the problem is, only that it exist a numerical error somewhere in the model.

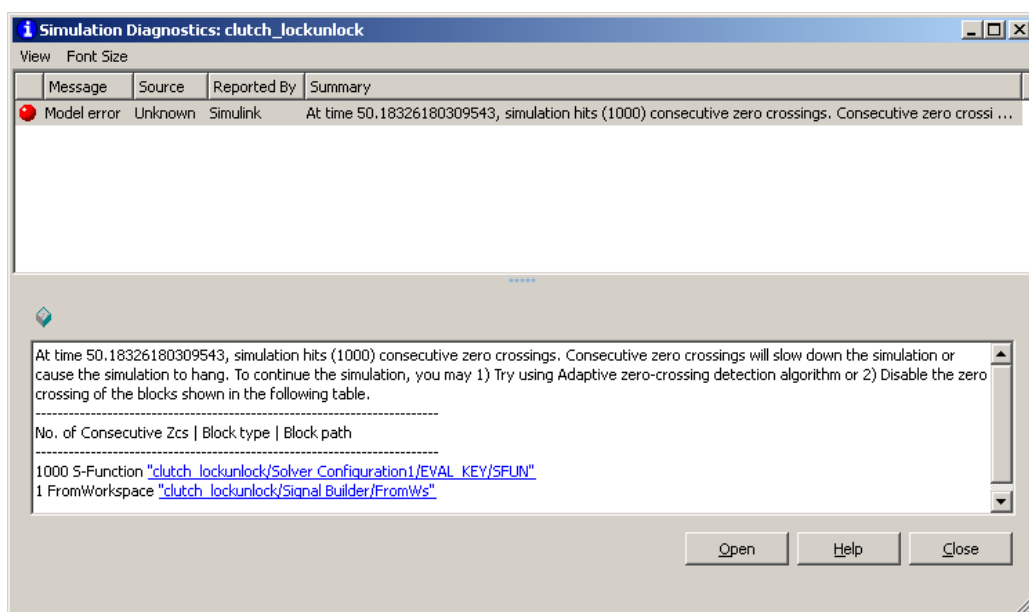


Figure 10: Zero crossing numerical error

- Higher order derivatives

For example it is not possible to write  $\phi.der.der == w$  directly. To use higher order derivative Simscape code 8 example approach must be used instead

**Simscape code 8: Higher order derivative approach**

```
variables
    h = { 0, 'rad/s*s' };
    w = { 0, 'rad/s' };
    phi = { 0, 'rad' };
end

equations
    phi.der == w;
    w.der == h;
end
```

## 6 Modeling of the ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C)

The gearbox that will be modeled in Simscape is the ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C). In this chapter the construction of the different components in Simscape are described in detail. It should be noted that the goal is to see the modeling potential of Simscape, rather than to model the gearbox in high fidelity.

The model will only be built of components that will fit in the mechanical rotational domain and they will all share the same variables, torque and angular velocity. As Figure 11 shows the following Simscape components need to be built:

- A Clutch model (included in the Torque Converter and the Gearset and Shift Mechanism)
- A Torque Converter (with lockup-clutch) model
- A Retarder model
- A Planetary Gear Train model (Gearset and Shift Mechanism)
- An Automatic Logic program (Transmission Control Unit)

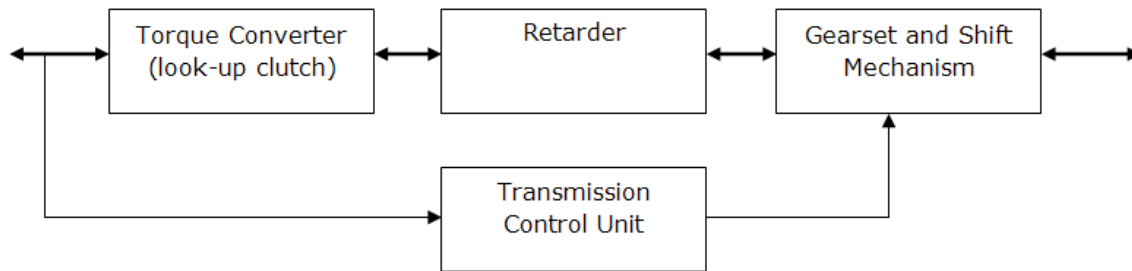


Figure 11: Model overview, the bidirectional arrows represents the physical network approach where the power can flow in both directions. The unidirectional represents one way signals.

### 6.1 The Clutch model

A clutch (or brake) can be hard to model realistically since the behavior of a clutch is usually modeled in a way where the clutch constantly shifts between continuous and discrete time events and this can be a difficult problem for many simulation environment solvers to handle. The clutch model has a certain importance for the gearbox model as a whole, because the clutch gets the dynamics working in a correct manner by locking and unlocking the planetary wheels, which in turn sets the direction of the power flow in the gearbox and that sets the different gear ratios. This will be described later in this chapter.

The clutch model in this chapter is based on a simplified version of real friction clutch that can have three different states and the only friction that will be accounted for is the dynamic (kinetic) friction, when the clutch plates are sliding.

To model the clutch, the following variables are introduced:

$\tau$ = Torque trough	[Nm]
$\tau_I$ = Torque input side	[Nm]
$\tau_O$ = Torque output side	[Nm]
$\omega = \text{abs} \omega_I - \omega_O $ , angular velocity	[rad/s]
$\omega_{tol}$ = Angular velocity tolerance	[rad/s]
$\omega_I$ = Angular velocity input side	[rad/s]
$\omega_O$ = Angular velocity output side	[rad/s]
$R$ = Radius of clutch plates	[m]
$A$ = Clutch area	[m]
$P$ = Clutch pressure	[kg/(m * s <sup>2</sup> )]
$F_f$ = Friction force	[(kg * m)/s <sup>2</sup> ]
$\mu_k$ = Kinetic friction constant	[no unit]
$N$ = Number of friction plates	[no unit]

The clutch torque capacity is given by:

$$\begin{aligned}\tau &= \iint_A \frac{R \times F_f}{A} dA [F_f = PA] = \iint_A \frac{R \times PA}{A} dA \left[ \int dA = \iint R dR d\theta \right] \\ &= P\mu_k \int_0^R \int_0^{2\pi} R^2 dR d\theta = P\mu_k \left( \frac{R^3}{3} \right) (2\pi)\end{aligned}\quad (1.1)$$

After some rearrangements the final torque equation yields:

$$\tau = \frac{2}{3} RP\mu_k A \quad (1.2)$$

The clutch area is thereafter expressed with the following formula, where the number of clutch plates is included:

$$A = \pi R^2 N \quad (1.3)$$

Next are the three different states that the clutch can switch between introduced:

- When the clutch is sliding and kinetic friction torque is transferred. The friction plates have different angular velocities and the angular velocity is above the threshold value. The sign function is the mathematical sign function and describe that the torque should act in the direction that opposes the slip.

$$\omega_I \neq \omega_O \quad (1.4)$$

$$\omega > \omega_{tol} \quad (1.5)$$

$$\tau = \frac{2}{3} RP\mu_k A * \text{sign}(\omega) \quad (1.6)$$

- When the clutch is sliding, but less torque is transferred. The friction plates still have different angular velocities, but  $w$  is approaching zero. The angular velocity is below the threshold value.

$$\omega_I \neq \omega_O \quad (1.7)$$

$$\omega < \omega_{tol} \quad (1.8)$$

$$\tau = \frac{2}{3} RP \mu_k A * \text{sign}(\omega) * \frac{\omega}{\omega_{tol}} \quad (1.9)$$

- When the clutch is locked. The two plates rotate together with the same angular velocities and the resulting equations become trivial.

$$\omega_I = \omega_O \quad (1.10)$$

$$\tau_I = \tau_O \quad (1.11)$$

The clutch states can thereafter be described in Simscape with the following piece of code:

#### Simscape code 9: Clutch equations

```
Equations
if  ((w>=w_tol) || (w<=-w_tol))    %Dynamic friction (SLIDING)

    t==2/3*R*P*Uk*A*sign(w);

else

    t==2/3*R*P*Uk*A*sign(w)*(w/w_tol);

end
end
```

The first thing to notice in the code is the first part of the equations is the kinetic friction only acts in a certain tolerance level, when  $\omega \geq \omega_{tol}$  or when  $\omega \leq -\omega_{tol}$ . This implementation is to avoid the discontinuing states (zero crossing) when  $\omega \rightarrow 0$ .

The zero crossing avoidance can be seen in the second part in the else statement  $\frac{\omega}{\omega_{tol}}$ , when the clutch is sliding.  $\omega$  in this case is the relative absolute value of the angular velocity which is measured on each side of the clutch, i.e.  $\omega = \text{abs}|\omega_I - \omega_O|$ .

The  $\text{sign}(\omega)$  is to express that the transferred torque should act in the opposite torque direction.

When  $\omega = 0$  the whole expression becomes zero which stands for the locked mode. The left side of the clutch's driveline is connected to its rights side and in this mode the both have the same angular velocity.

$$\omega_I = \omega_O \quad (1.12)$$

$$T_I = T_O \quad (1.13)$$

The clutch will stay in this locked mode as long as the pressure  $P$  is enough. When  $P$  decreases the reverse order is applied and when the pressure reaches zero the clutch's left and right side of the driveline is now completely separated and can rotate independently from each other.

## 6.2 The Torque Converter model

It is very difficult to construct a representable physical model of a torque converter since it represents a fluid coupling and therefore knowledge about the physics of the flow have to be known in every single state, which often leads to lengthy and cumbersome equation. The full dynamic perspective of the converter is therefore hard to model analytically, which is not in the scope for this thesis. An easier way, instead of trying to describe it physically, is to describe the power conversion with mathematical formulas which are based on measurement data from the specific torque converter.

In doing so, the behavior of the specific flow can be expressed implicitly and a good representation of the physical torque converter can therefore be reached. The technical data about the behavior of the torque converter is taken from the technical manual provided by ZF Friedrichshafen AG.

To describe the performance of the converter the following variables are introduced:

$\tau_I$ = Torque impeller side	[Nm]
$\tau_T$ = Torque turbine side	[Nm]
$\omega_{min}$ = Angular min value	[rad/s]
$\omega_I$ = Angular velocity impeller side	[rad/s]
$\omega_T$ = Angular velocity turbine side	[rad/s]
$D$ = Torque Converter diameter	[m]
$\rho$ = Density of medium	[(kg/m <sup>3</sup> )]
$R_\omega$ = Speed ratio	[no unit]
$K$ = Capacity factor	[no unit]
$R_\tau$ = Torque ratio	[no unit]

First is the function that specifies the relation between input (impeller) and the output (turbine) speed introduced.

$$R_\omega = \frac{\omega_T}{\omega_I} \quad (1.14)$$

Thereafter are the two functions that specify the characteristics of the converter: the torque ratio  $R_\tau$  and the capacity factor  $K$ , both as functions of the speed ratio  $R_\omega$ .

$$R_\tau = f(R_\omega) \quad (1.15)$$

$$K = f(R_\omega) \quad (1.16)$$

Torque conversion ratio is a term used to express the ratio between the impellers torque and the turbine torque. The greater the speed difference between the impeller and turbine, the greater the ratio is. Hence, it multiplies the engine torque at a low engine speed when the difference is the greatest. This improves the vehicle's launching performance and responsiveness. As the speed of the turbine wheel increase, the torque conversion falls. The capacity factor is dependent on the detailed geometry regarding blade angles, fluid density and viscosity.

In normal operation the torque on both the input impeller and output turbine can be expressed with the following two equations which are presented in (BOSCH, 2007).

$$T_I = KD^5 \omega_I^2 \quad (1.17)$$

$$T_T = T_I * R_T \quad (1.18)$$

The final torque converter Simscape model is described with the following piece of Simscape code:

**Simscape code 10: Torque Converter equations**

```
equations
R_w == (w_T / w_I) ;

if ( ((w_I >= w_min) || (w_I <= -w_min) || (w_T >= w_min) || (w_T <= -w_min)) )

    T_I == K * D^5 * w_I^2 * p * (1 - (w_T / w_I)) ;
    T_T == - T_I * R_T ;

else

    T_I == K * D^5 * w_min^2 * p * (1 - (w_T / w_I)) ;
    T_T == - T_I * R_T ;

end
end
```

As in the clutch model, a  $\omega_{\min}$  variable is introduced to avoid the zero crossing when the simulation starts and either  $\omega_I$  or  $\omega_T$  can be 0. The  $(1 - \frac{\omega_T}{\omega_I})$  function in equation  $\tau_I$  is to express that the torque converter should not transfer any torque when the lock-up clutch is active. When  $\omega_I = \omega_T$ , the expression  $(1 - \frac{\omega_T}{\omega_I})$  becomes zero.



will be direct dependent on the gear engagement. The highest braking levels are available in the lower gears, and therefore at lower vehicle speeds.

The following variables are introduced:

$\tau_I$ = Torque input	[Nm]
$\tau_O$ = Torque output	[Nm]
$\tau_R$ = Retarder torque output	[Nm]
$\omega_p$ = Angular velocity propeller shaft	[rad/s]
$R$ = Current gear ratio	[no unit]
$P$ = Percentage ratio	[no unit]
$E$ = Engine friction ratio	[no unit]
$R_\tau$ = Retarder ratio	[no unit]

First there is one function that specifies the characteristics of the retarder, the retarder ratio as a function propeller shaft speed (output from gearbox):

$$R_\tau = f(\omega_p) \quad (1.21)$$

In normal operation the decreasing torque transmitted when the retarder is active is described with the following equation:

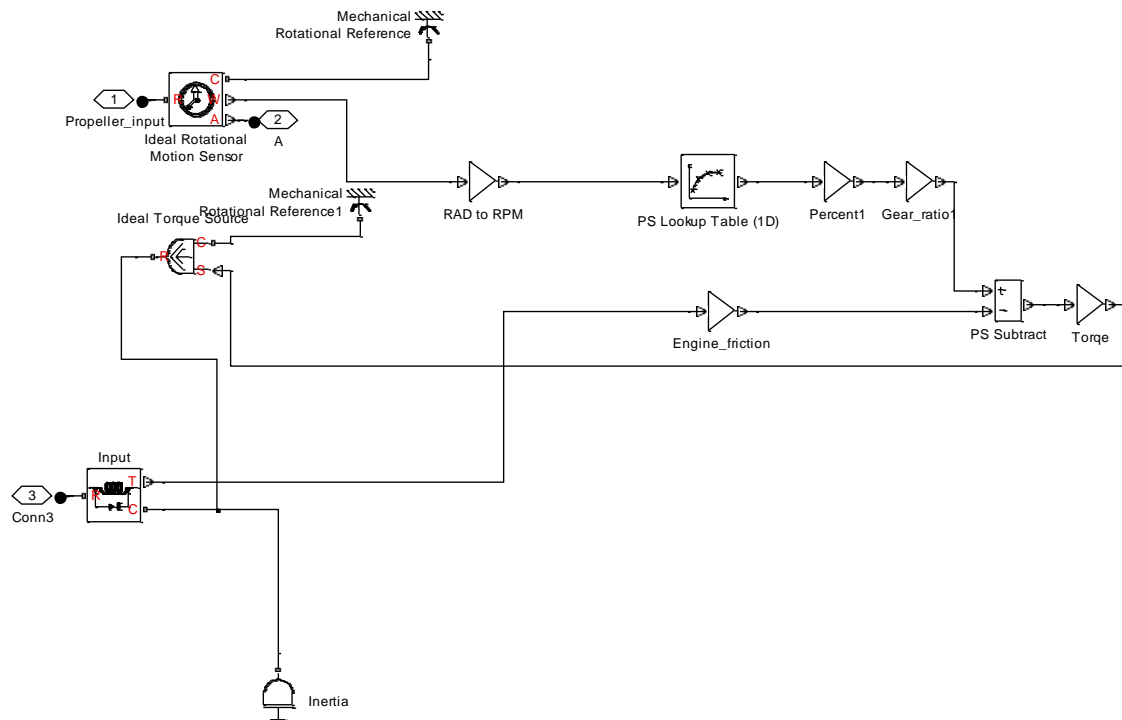
$$\tau_R = \frac{R_t P}{R} - \tau_I E \quad (1.22)$$

The final equation for the retarder results in:

$$\tau_O = \tau_I - \tau_R \quad (1.23)$$

In the parameters described above, there are some values that are fixed in the model, but in a more realistic model in the future they could easily be changed to variable parameters. The engine friction parameter is set at constant value of 0.1 which is a quite reasonable measure of engine friction. The percentage ratio is set to 0.8; this ratio is in the reality dependent on number of different factors such as: requested braking torque from the driver, brake pedal position. The gear ratio is set to a fixed value 1.6 but should in a more realistic model have a variable value that changes according to the current gear ratio.

The above equations results in the following Simscape model seen in Figure 13, notice the use of Simscape physical signals.



**Figure 13: The retarder model**

## 6.4 The Planetary Gear Train model

This is the central part of the automatic gearbox and this is mainly where the angular motion and torque conversion mechanism is taking place. Modeling of this part is really taking advantage of the physical network approach. In the technical manual of the ZF-ECOMAT 4 series there is a diagram over the power flow through the gearbox when the different gears are active which is ideal for modeling with this approach. The schedule shows which clutches are active, which are not and how the information flow is directed through the three different planetary gears which will be described below.

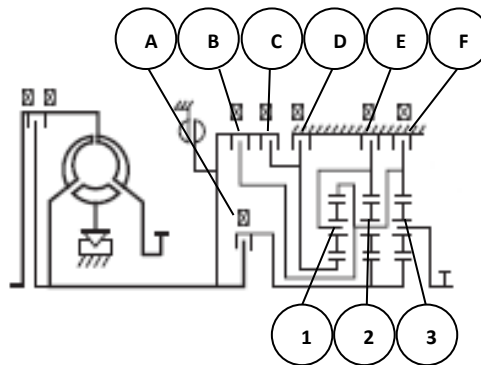


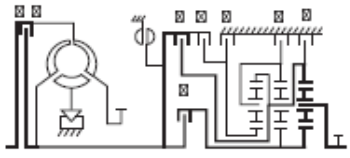
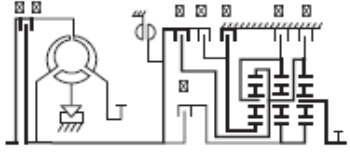
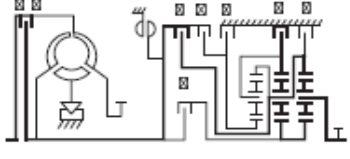
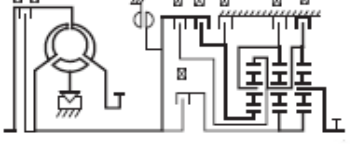
Figure 14: Power flow schedule

Figure 14 above illustrates the inside of the gearbox and how the power can flow through the gearbox. The lettered rings represent the different clutches in the gearbox and the numbered rings represent the three planetary gears.

In Table 6 below the power flow diagrams are shown for each gear, where the bold selection illustrates the power flow.

Table 6: Power flow diagram for each gear

Clutches active	Power flow
(A, F)	1 <sup>st</sup> gear
(A, E)	2 <sup>nd</sup> gear
(A, D)	3 <sup>rd</sup> gear

(A, B)	4 <sup>th</sup> gear	
(B, D)	5 <sup>th</sup> gear	
(B, E)	6 <sup>th</sup> gear	
(C, F)	R-gear	

Since a model of a clutch already has been built the next components that will be modeled is the planetary gear.

#### 6.4.1 Planetary gear sets

The planetary gears can as described in the *Chapter 3.2.4 The Planetary Gear Sets* shift power conversion depending on which axle acting as a fixed one, and which one is acting as the input. The first one who gave a detailed explanation of the physics of the planetary gears was Robert Willis in his book *Principles of Mechanism* 1870. (WILLIS, Robert, 1870)

The following variables have to be introduced to describe the planetary gear:

$\tau_S$ = Torque sun	[Nm]
$\tau_C$ = Torque carrier	[Nm]
$\tau_R$ = Torque ring	[Nm]
$\omega_S$ = Angular velocity sun	[rad/s]
$\omega_C$ = Angular velocity carrier	[rad/s]
$\omega_R$ = Angular velocity ring	[rad/s]
$Z_S$ = Number of teeth sun wheel	[no unit]
$Z_R$ = Number of teeth ring wheel	[no unit]

Since planetary gears have three connecting ports that they share with the physical network, three different equations must be derived to describe the relationships between the components nodes and variables.

The first one is the basic kinematic equation for planetary gear sets:

$$\omega_S + \left(\frac{Z_R}{Z_S}\right) * \omega_R - \left[1 + \frac{Z_R}{Z_S}\right] * \omega_C = 0 \quad (1.24)$$

The last two equations can be derived from doing a torque analysis. The energy balance equation for planetary gear train can be written as:

$$\tau_R \omega_R + \tau_C \omega_C + \tau_S \omega_S = 0 \quad (1.25)$$

And when the planetary gear is moving along a solid axle the angular speed is the same ( $\omega_R = \omega_C = \omega_S$  and  $\omega \neq 0$ ). The equation above can now be written as:

$$\tau_R + \tau_C + \tau_S = 0 \quad (1.26)$$

*Equation 1.26* is the first energy equation that defines the different torque conversions taking place. The second one is when the carrier is held stationary ( $\omega_C = 0$ ). *Equation 1.24* becomes now:

$$\frac{Z_R}{Z_S} * \omega_R = -\omega_S \quad (1.27)$$

Together with the energy balance *Equation 1.25* and once again setting  $\omega_C = 0$  the resulting equation is, after some rearrangement:

$$\tau_R = \frac{Z_R}{Z_S} * \tau_S \quad (1.28)$$

The three *Equations, (1.24) (1.26) and (1.28)* that characterize an ideal planetary gear have now been derived. When the mathematical representation is done it can easily be transformed into a planetary gear component through some programming in the Simscape language. It could look like the code piece below:

**Simscape code 11: Planetary gear equations**

```
equations

    %Overall kinematics
    (1 + (Z_R/Z_S)) * w_C == w_S + (Z_R/Z_S) * w_R;

    %Torque balance
    T_R == (Z_R/Z_S) * T_S;
    0 == T_C + T_S + T_R;

end
```

#### 6.4.2 The full gear model

When the two components have been built it is time to transfer the power flow diagram into a Simscape model which can be seen in the Figure 15 below. The model is a representation of the power flow diagram and it can simulate how the actual power flows through the gearbox.

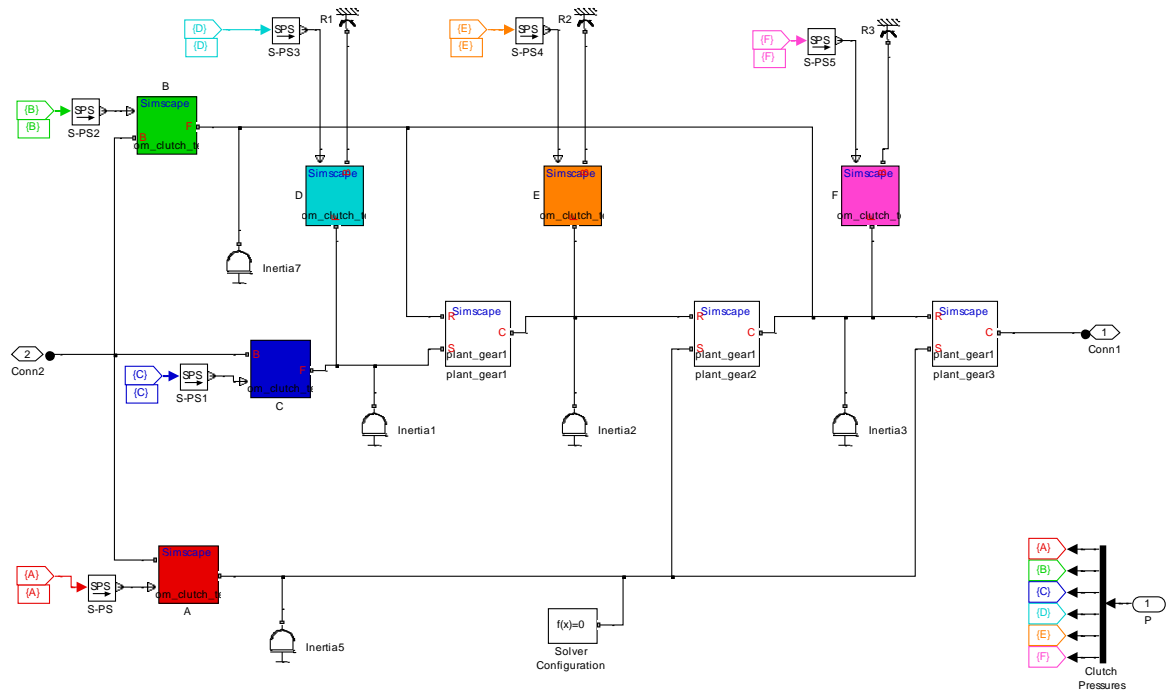


Figure 15: The full gear model

### 6.4.3 Gear Ratios

The different gear ratios for the ZF-ECOMAT 4 are presented in Table 7 below:

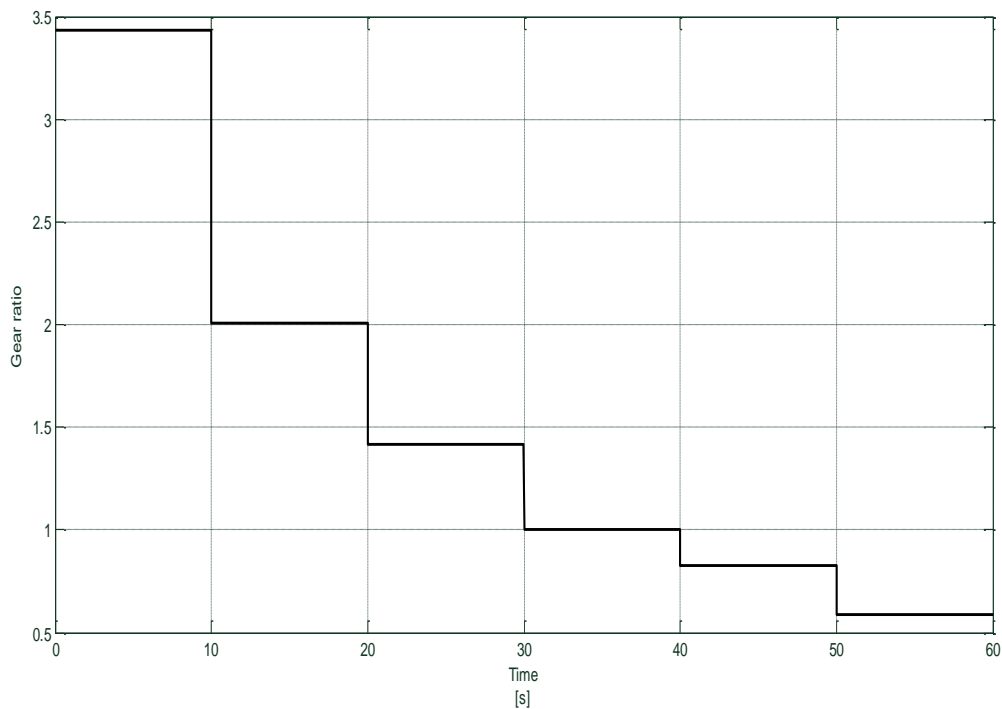
Table 7: ZF-ECOMAT 4 ratios, (ZF FRIEDRICHSHAFEN AG , 2006)

Gear	1	2	3	4	5	6
Ratio	3.43	2.01	1.42	1.00	0.83	0.59

In order to get each corresponding ratio correct the ring to sun ratio  $\left(\frac{Z_R}{Z_S}\right)$  has to be adjusted for each of the three planetary. The ratio parameters have been incrementally adjusted until it could match the ZF-ratios. The ratio adjustment  $\left(\frac{Z_R}{Z_S}\right)$  for each planetary gear presented in Table 8 below and the result of the ratio adjustment can be seen in Figure 16 on the next page.

Table 8: Ratio adjustment

Planetary Gear 1 (ring to sun)	Planetary Gear 2 (ring to sun)	Planetary Gear 3 (ring to sun)
2.405	2.4396	2.43



**Figure 16: Result of the adjusted gear ratios for gear 1-6**

## 6.5 The Automatic Logic program

The control logic for the gearbox is implemented with help of another tool made by MathWorks called Stateflow. Stateflow enables a graphical representation for event-driven systems based on finite-states. Gear shift control algorithms in many modern automatic gearboxes can be extremely complex and often takes many years to develop. Usual parameters that these algorithms take account for when shifting gears are for example throttle position, engine rpm, gps data, current gear, incline, torque request etc. The control logic for this model will have a more simple character and will only have one input parameter and that is the engine angular velocity. Below is the actual Stateflow chart shown which is based on a modified version of one of the examples provided with Stateflow called sf\_car.

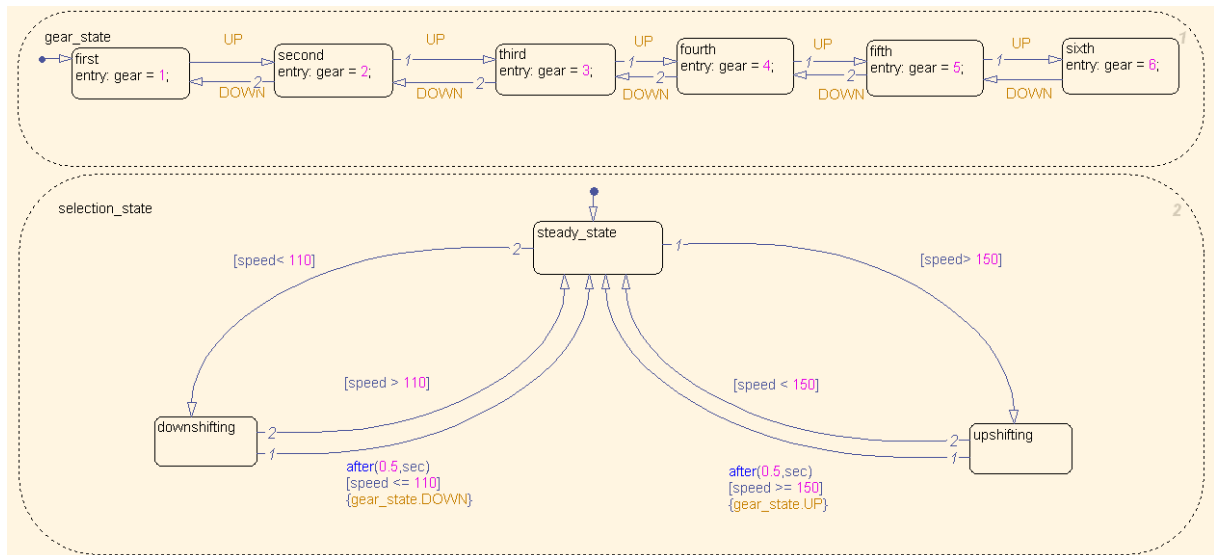


Figure 17: Automatic logic Stateflow chart

The gear shifting logic is very simple to follow.

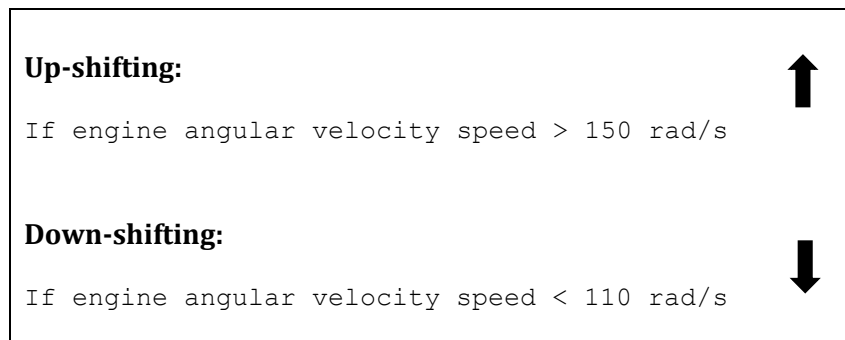


Figure 18: Gear shift logic

One thing to mention though is that both up-shifting and down-shifting can only take place if the after trigger is fulfilled. The after trigger represents the time delay it takes before shifting can occur. In the model it is set to 0.5 seconds.

## 7 Validation

As mentioned earlier it is trivial to build accurate models, the difficulty lies in to build models with sufficient accuracy for the purpose of the study in order to give the results credibility. It is an interesting question to ask how close this simplified gearbox model follows the behavior of a real system and since no separate component validation has been carried out.

Therefore, the validation process will be carried out in the following way:

The Simscape model will be tested against given signals recorded from a measurement done on Scania's test track in Södertälje. The measurement was done in King-Kong which is one of Scania's test busses.

The bus was accelerated to a speed of 80 km/h and thereafter it was freewheeling for a couple of seconds. The total measurement time was 35 s and the following three signals were recorded:

- Torque output from the engine (flywheel torque)
- Speed on the bus (tire speed)
- Current gear

The test vehicle was at the measurement equipped with the following components seen in Table 9 below:

**Table 9: List of test bus specifications**

<b>Component</b>	<b>Specification</b>	<b>Type</b>
Vehicle	Double-Decker (King-Kong), Mass 11 440 kg	Scania KUD 4*2
Engine	9 liter, Maxpower 206 kW (280 Hp), Maximum Torque 1400 Nm	Scania DC 929
Gearbox	6 speed gearbox	ZF ECOMAT HP 604 C
Final Gear	Ratio 6,2	RD 760
Tire	Radius 0,507 m	295/80

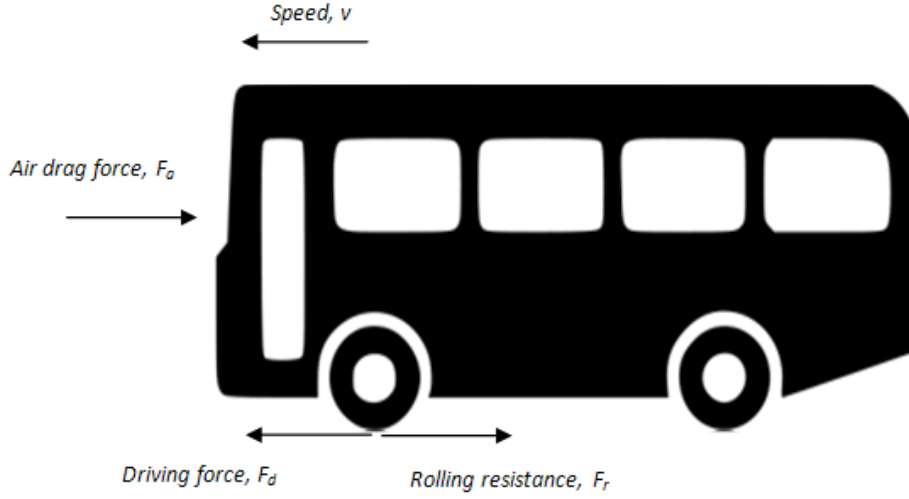
### 7.1 Remaining approximations

Because the model, as it has been described to this point, only represents the gearbox system (torque converter, retarder, gear sets) and this validation process will compare the recorded speed of the actual bus against the speed of the model; some approximations have to be made for the remainder of the driveline.

The following approximations are therefore made:

After the gearbox comes a final gear with a value set to 6.2 which correspond to King-Kong's value.

The approximation of the longitudinal forces acting on a bus with *mass*  $m$  and *speed*  $v$  on a flat surface can be seen in the Figure 18 on the next page.



**Figure 19: Approximation of longitudinal forces acting on a bus**

Newton's second law in the longitudinal direction states:

$$m\bar{a} = \bar{F}_d + \bar{F}_a + \bar{F}_r \quad (1.29)$$

If positive direction is defined to the left in Figure 19 above, Equation 1.29 gives:

$$ma = F_d - F_a - F_r \quad (1.30)$$

The air drag can be expressed with:

$$F_a = \frac{1}{2} c_w A_a \rho_a v^2 \quad (1.31)$$

Where,  $c_w$  is the drag coefficient,  $A_a$  the maximum vehicle cross section area, and  $\rho_a$  the air density

Rolling resistance:

$$F_r = (C_{r1} + C_{r2}v) * \frac{m}{1000} \quad (1.32)$$

Where,  $C_{r1}$  and  $C_{r2}$  are rolling resistance coefficients depending on tire and tire pressure expressed in tons, and that is why the expression is divided by 1000.

Further, Equation 1.30 together with  $v_{vehicle} = \omega_{vehicle} * r_{tire}$  yields the resulting driveline equation:

$$\omega r = \int \frac{F_d - F_a - F_r}{m} \quad (1.33)$$

In this stage an approximated complete vehicle model has been created in which a given torque input signal gives a roughly approximated vehicle output speed. The final validation concept can be seen in the Figure 20 below and the corresponding Simscape model can be found in Appendix 12.5 The final validation concept. In the figure is a torque actuator component which in this case represents the actual engine and it ignites the system with the measured torque signal from the engine.

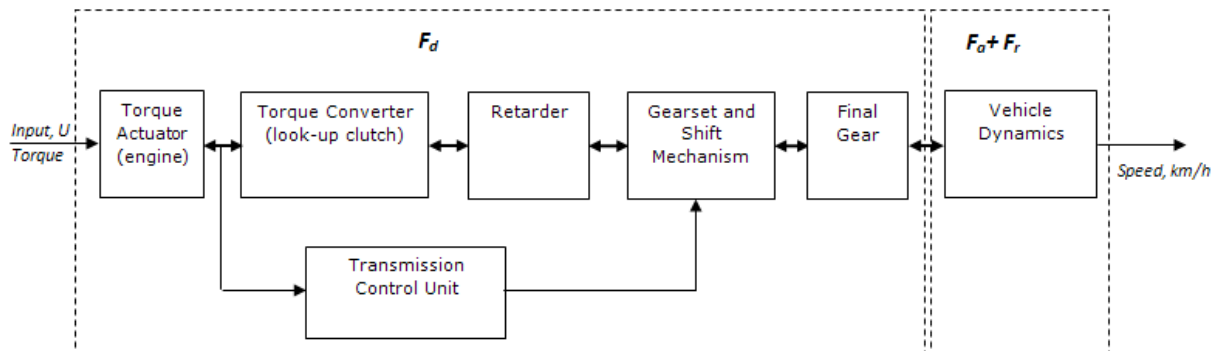


Figure 20: Final validation concept, the bidirectional arrows represents the physical network approach where the power can flow in both directions. The unidirectional represents one way signals.

The measured flywheel torque signal that will be used as the input signal to the model can be seen in the figure below:

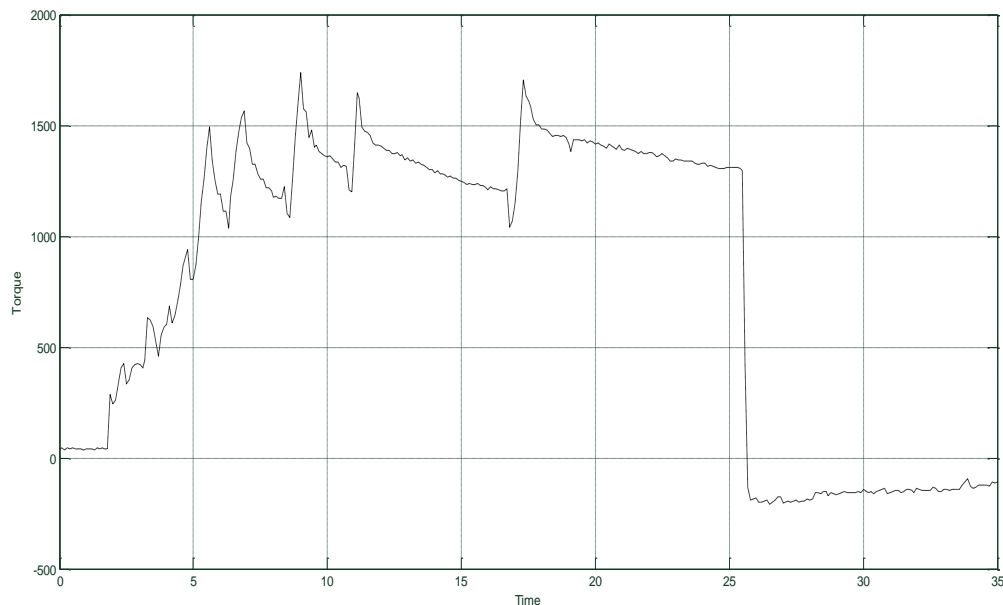


Figure 21: Torque input signal

## 8 Results

Two tests were carried out to verify the model. The input signal to the model was the measured torque signal (see Figure 20). The differences between the two tests were the gear shift strategy. The first one uses the automatic logic described earlier in the modeling chapter and the second one uses the actual recorded gear shift signal as input. Figure 23 illustrates the gear shift signal when Stateflow logic is active and Figure 26 the recorded gear shift signal. The speed results of the two tests carried out can be seen in Figure 22 and 25 and the absolute error diagrams can be seen in Figure 24 and 27.

### 8.1 Test 1, with the automatic shift logic:

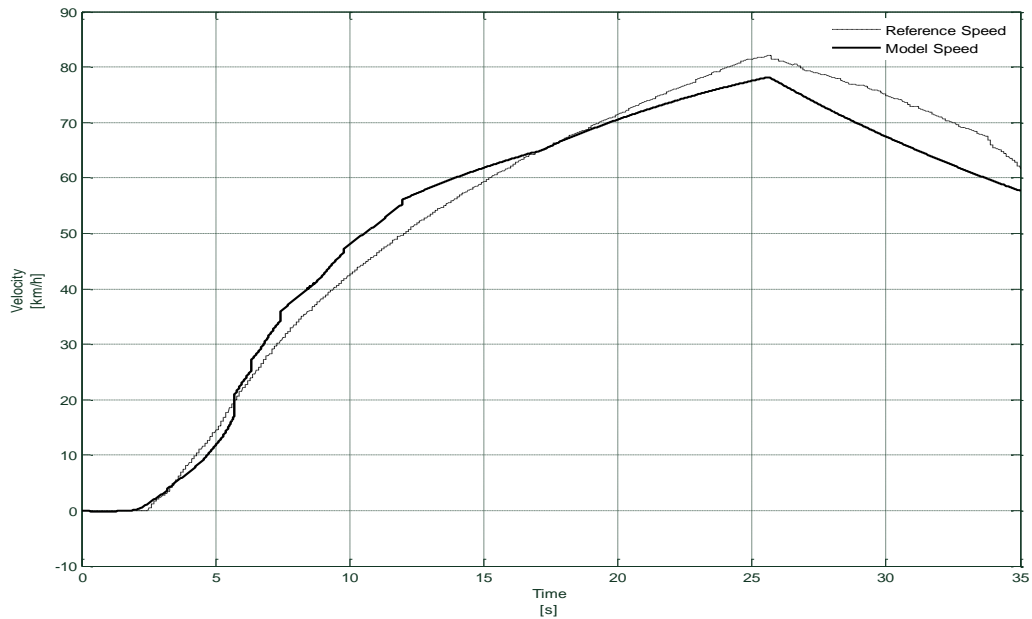
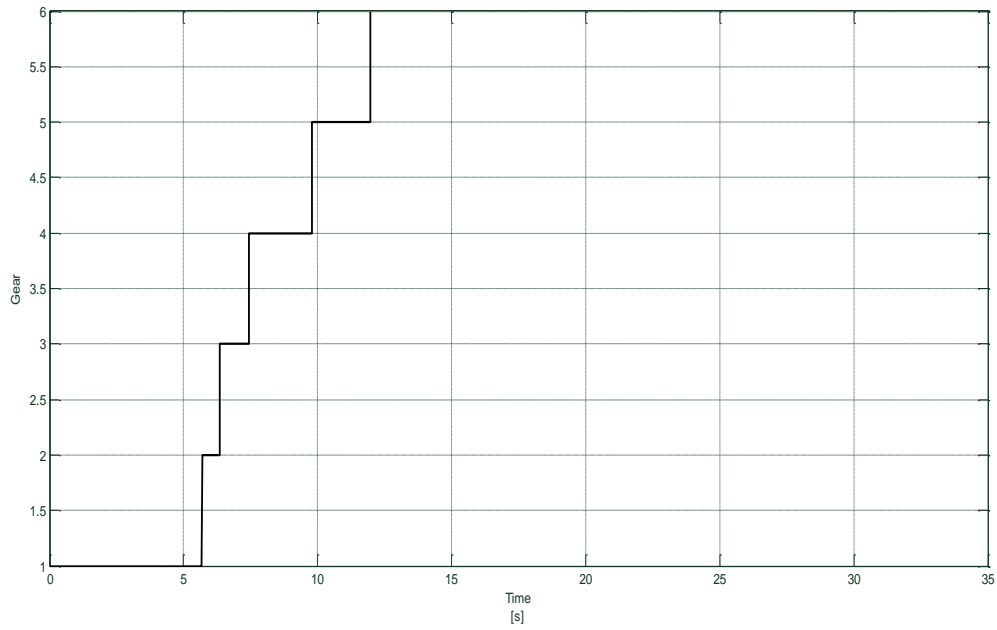
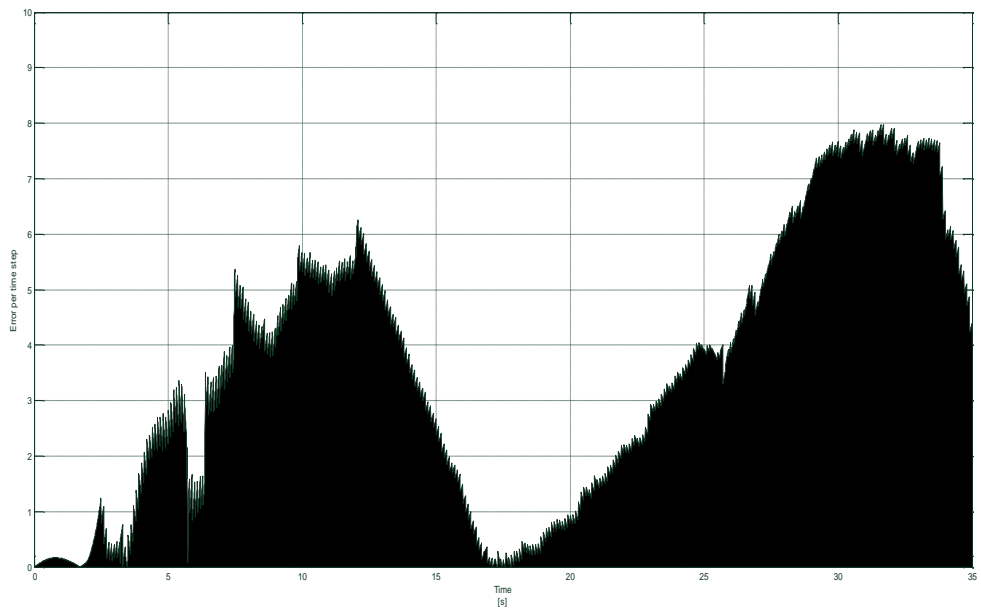


Figure 22: Test 1, Model speed versus reference speed



**Figure 23: Test 1, Gear shifts with Stateflow**



**Figure 24: Test 1, Absolute error between model and reference speed in [km/h],  $\text{abs}(V_{\text{reference}} - V_{\text{model}})$ . A trapezoidal numerical integration gives the total error sum of 124.6099 area units.**

## 8.2 Test 2, with the recorded gear shift signal:

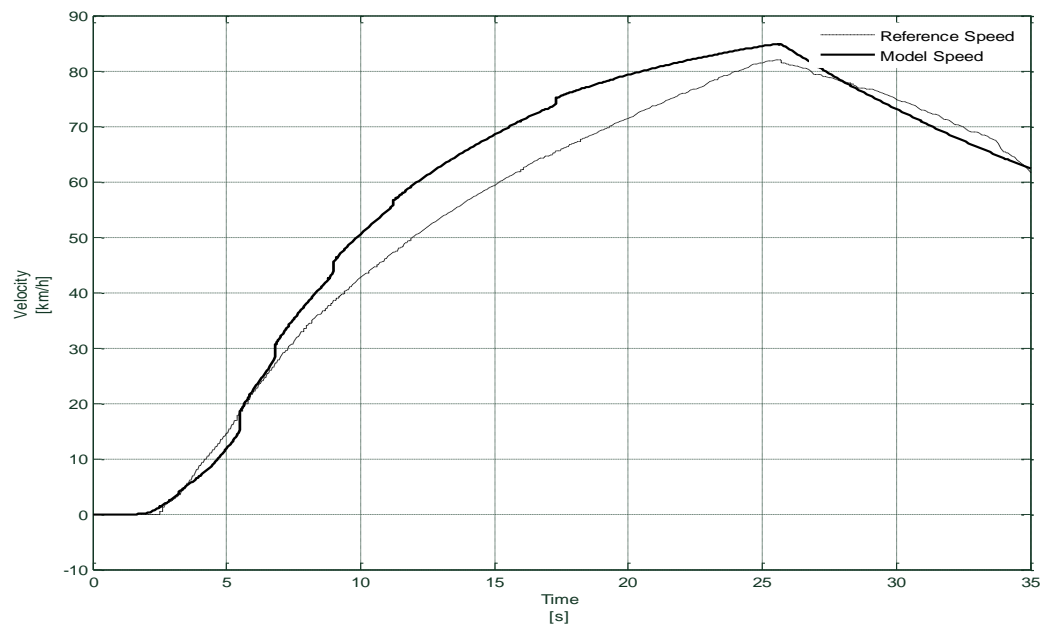


Figure 25: Test 2, Model speed versus reference speed

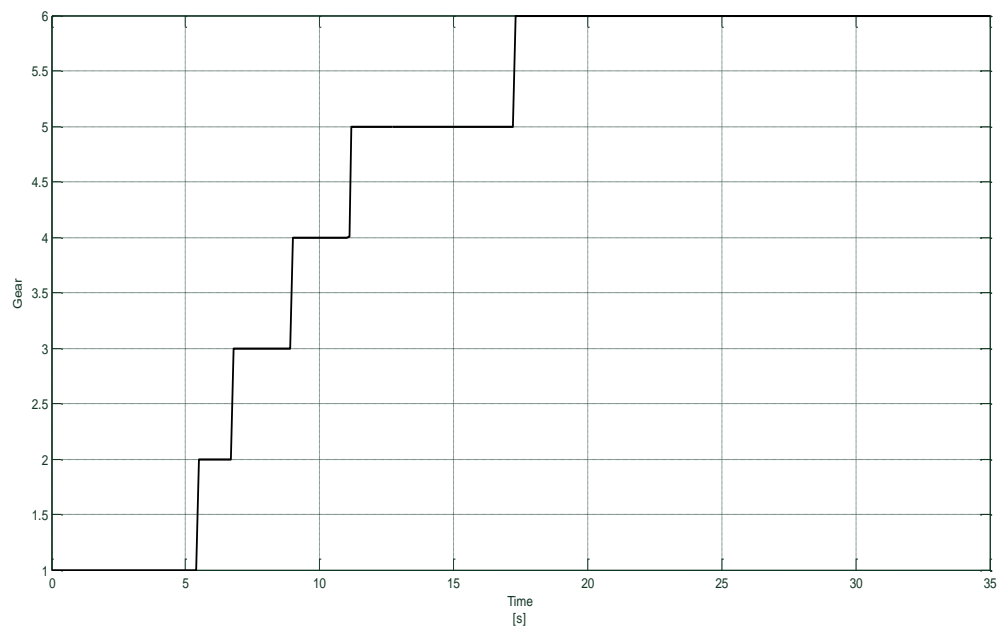
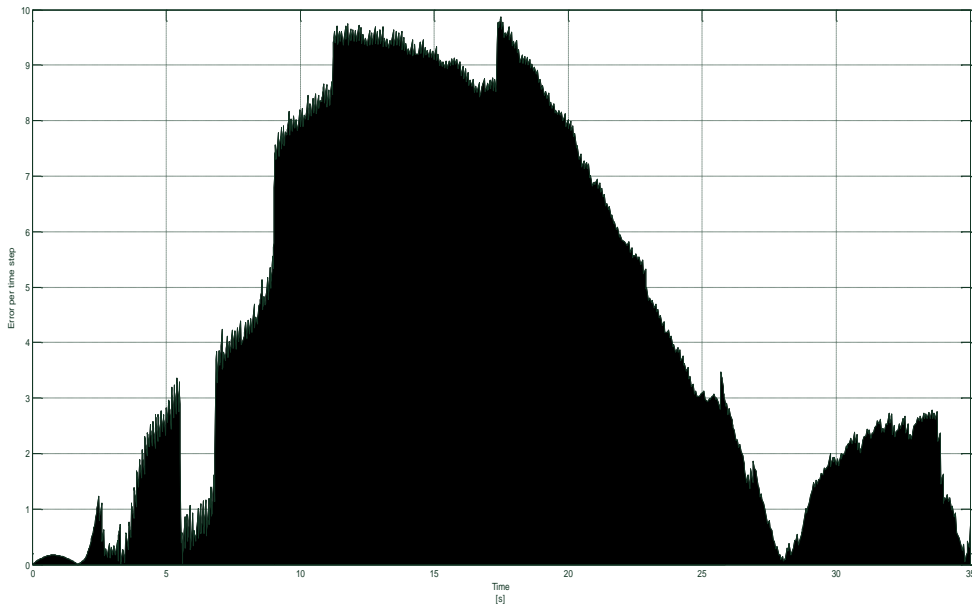


Figure 26: Test 2, Recorded gear shift signal



**Figure 27: Test 2, Absolute error between model and reference speed in [km/h] ,  $\text{abs}(V_{\text{reference}} - V_{\text{model}})$ . A trapezoidal numerical integration gives the total error sum of 158.3533 area units.**

### 8.3 Results discussion

The purpose of this thesis was to analyze the feasibility and usability for modeling with MathWorks simulation tool Simscape by building a simplified model of the automatic gearbox ZF-ECOMAT 4 (HP 504 C / HP 594 C / HP 604 C). It has been shown throughout the thesis how this model was built. First the system knowledge has been acquired by studying relevant literature and speaking with the persons concerned. The second step was to get acquainted with Simscape and the physical network approach. The physical network approach that is accessible through the Simscape language makes it easy to build custom made components with means of physical and mathematical relationships. With this background a stepwise approach has been conducted which has led to the final model of the gearbox and the validation concept.

The results of the two tests indicate that the dynamic is working in a correct manner. In both tests the model speed follows the reference speed quite accordingly through the different gears. The fact that test 1 gave a less total speed margin of error than test 2 (124.6099 km/h compared to 158.3533 km/h), which also can be seen in the Figure 24 and 27 is only due to the difference in gear shifts. The Stateflow automatic logic program for test 1 gave the model a more moderate speed curve while the recorded gear shift signal for test 2 gave the model a steeper speed curve. One thing to notice from both of speed figures 22 and 25 is that when a gear change is taking place a rapid speed jump can be seen which takes place in a very small timeframe.

The two tests show that even though the model is simplified it still is reasonably accurate. In a more realistic vehicle model the difference between the two tests would probably rather result in the opposite if the current shift logic program is kept as it is now. The fact that test 1 gave a less total speed margin of error is more of a coincidence and depends probably on the

construction of the simplified vehicle model or incorrect parameter settings when testing the model.

The rapid speed jumps that takes place when shifting gears is due to the powerful torque transfer trough the clutches when acting in slip mode ( $\omega_I \neq \omega_O$ ). This can be seen in Appendix 12.4 Torque output on propeller shaft, where the highest peak transfer reaches appriximately 12000 *Nm* during 0.1 seconds. This behaviour is a result of the parameters used in the clutch model. Most impact for the two tests has the pressure parameter which actuates the clutch. For the two tests all clutches are actuated with a value of 45000 *Pa*. In a more realistic clutch model that value may have to be adjusted and that will decrease theese jumps and make the gear shifting time frame last for a longer period. A more realistic engine model will also consider gear shifting factors such as engine rpm syncing, and decrease in torque before shifting which will further reduce theese jumps.

One thing that can be mentioned is also that no breaking simulations have been carried out because of the time frame requirements for the thesis. Therefore, the retarder component has been inactive in the whole validation process.

Nevertheless, this initial model can serve as good ground for further development and gives a good indication that it is possible to build and simulate realistic physical models with the Simscape software. How realistically is hard to tell solely from the resulsrts of this thesis work and need further studies.

## 9 Conclusion

Simscape has proven to be a powerful tool for modeling physical systems and all of the demands that Ljung & Glad listed that should be fulfilled for a modern simulation tool has more or less been met which can be seen in Table 10.

**Table 10: Demands for a modern modeling and simulation tool**

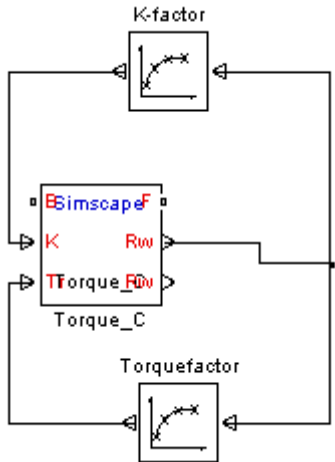
<b>Demand</b>	<b>Fulfilled</b>
It should cover as many physical and technical domains as possible.	✓
It should be systematic. Ideally, large parts should be automated in the software.	✓
It should lead to a mathematical formulation that is appropriate for simulation and other modeling uses.	✓
It should be modular. It should therefore be possible to build component models that can then be assembled into complete systems.	✓
It should facilitate the reuse of models in the new context	✓
It should be close to the physics. It must therefore resemble the real physical world in an accessible way.	✓

However, with that said there are still some things that must be improved to make the tool more accessible, mainly from a usability perspective. Below is a list of suggestions that I suggest could make Simscape a lot more accessible, effective and decrease the learning curve, especially for inexperienced users of physical modeling. First comes the suggestions which I think are the most important and thereafter are the less important.

### Important

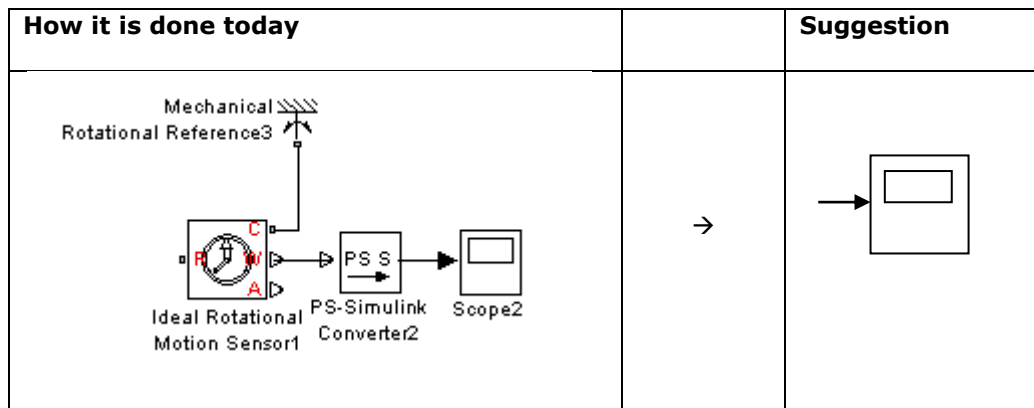
- A larger library of included components for all of the provided physical domains with included source code and documentation.
- More examples of different models build in respective physical domain, which the user can study and test. Both simple and more complex models.
- A better troubleshooting guidance would make the modeling a lot more efficient. If it exists numerical issues in the model the error message does not say in which component the problem arise. The modeling approach has as mentioned before started simple by testing each component separate to eliminate some of these issues. Hence, when adding more components some numerical problems may still persists and then it becomes even harder to know in which component the problem exists if Simscape does not tell you specifically where or what the root cause of the problem is.
- The possibility to use more MATLAB defined functions and to make your own custom made functions.

For example, and also how it is done in the torque converter model is that lookup tables are taken from the physical signal library. If the MATLAB lookup table could be integrated as function in Simscape library no outgoing physical signals would be needed for the Simscape block which would increase the flexibility with the tool and make the models less cluttered.

How it is done today		Suggestion
	→	<pre> equations  T<sub>I</sub>==Interpolate((w<sub>T</sub>/w<sub>I</sub>),Kvalues)* D^5*w<sub>I</sub>^2*p*sign(1-(w<sub>T</sub>/w<sub>I</sub>));  T<sub>T</sub>== -T<sub>I</sub>*Interpolate((w<sub>T</sub>/w<sub>I</sub>), R<sub>T</sub>values);  end </pre>

## Less important

- When building models (even the simplest ones) they often become unnecessary cluttered. A common approach to reduce this problem is by using subsystems in order to be able to visualize the complete models. For example to look at the angular velocity; four different Simscape/Simulink components have to be used if they are not masked as a subsystem, where one component should be sufficient for this.



- Boolean operators would further extend the object-oriented possibilities with the Simscape language.
- Better scope features without having to go the detour via MATLAB, such as:
  - A scope that can look at Simscape physical signals directly, without the need of a PS-Simulink converter block
  - Export data other formats
  - Save scope data to a picture
  - Compare data to another data source, simulation
  - Change the visual appearance (colors, fonts, titles etc.)
- To be able to activate/deactivate a block. If two components should be tested one of them have to be deleted from the model before simulation start. If it could still exist in the model but exist as passive element it would make the testing process more efficient and it would become easier to test different components and configurations.

## 9.1 Recommendations for future work

One interesting field to look closer into would be to build a more complex gearbox model. My suggestion for extending the current ZF-ECOMAT 4-model would be the following:

- A more realistic clutch model, with a more realistic friction dynamics.
- A torque converter where fluid drag losses is included.
- Gearbox dynamic such as friction (bearing and mesh), elasticity and backlash.
- A more elastic driveline.
- Model the hydraulic system in the gearbox, maybe the clutches can be actuated with a hydraulic pressure source.

Further interesting topics for deeper Simscape studies would be to extend the gearbox model with more driveline components models and mix different physical domains. Could a more complete vehicle model match the reality better and what happens with the simulation time when the model complexity increases?

One limitation of this thesis is also that each model component has not been separately validated against a reference. Further studies would need to validate each component separately in a more accurate way, before merging them into a complete gearbox system.

## 10 Bibliography

Amazon.com. [online]. [Accessed 13 Oct 2010]. Available from World Wide Web:

<[http://www.amazon.com/Roger-Dean-Smith/e/B0032ONQEI/ref=ntt\\_dp\\_epwbk\\_0](http://www.amazon.com/Roger-Dean-Smith/e/B0032ONQEI/ref=ntt_dp_epwbk_0)>

BERGLUND, Niklas. 2010.

BOSCH. 2000. *Automotive Handbook 5th Edition*. Stuttgart: Robert Bosch, GmbH.

BOSCH. 2007. *Automotive Handbook 7th Edition*. Germany: John Wiley & Sons Ltd.

*Engineering Simulation Solutions for the automotive Industry*. 2008. ANSYS.

*European Automobile Industry Report*. 2009-2010. European Automobile Manufacturers Association.

LJUNG, Lennart and Torkel GLAD. 2008. *Modellbygge och simulering*. Lund: Studentlitteratur.

Mathworks.com--Recorded Webinar: *Physical Modeling with the Simscape Language*. [online].

[Accessed 06 Oct 2010]. Available from World Wide Web:

<<http://www.mathworks.com/company/events/webinars/wbnr49772.html?id=49772&p1=723914514&p2=723914532>>

MILLER, Steve. 2008. *Modeling Physical Systems as Physical Networks with the Simscape Language*. The MathWorks.

Scania.se. [online]. [Accessed 18 Jan 2011]. Available from World Wide Web:

<<http://www.scania.se/om-scania/scaniakoncernen/>>

*Simscape™ 3 Language Guide*. 2010. Natick: MathWorks.

*Simscape™ 3 User's Guide*. 2010. Natick: MathWorks.

SMITH, Roger D. 2003. Simulation. In: *Encyclopedia of Computer Science, 4th edition*, Chichester, UK: John Wiley and Sons Ltd, pp.1578 - 1587.

Wikipedia--Automatic\_transmission. [online]. [Accessed 08 Oct 2010]. Available from World Wide

Web: <[http://en.wikipedia.org/wiki/Automatic\\_transmission](http://en.wikipedia.org/wiki/Automatic_transmission)>

Wikipedia--Clutch. [online]. [Accessed 18 Oct 2010]. Available from World Wide Web:

<<http://en.wikipedia.org/wiki/Clutch>>

Wikipedia--List\_of\_ZF\_transmissions. [online]. [Accessed 30 Sep 2010]. Available from World Wide

Web: <[http://en.wikipedia.org/wiki/List\\_of\\_ZF\\_transmissions](http://en.wikipedia.org/wiki/List_of_ZF_transmissions)>

Wikipedia--Overdetermined\_system. [online]. [Accessed 28 Jan 2011]. Available from World Wide

Web: <[http://en.wikipedia.org/wiki/Overdetermined\\_system](http://en.wikipedia.org/wiki/Overdetermined_system)>

Wikipedia--Torque\_converter. [online]. [Accessed 30 Sep 2010]. Available from World Wide Web:

<[http://en.wikipedia.org/wiki/Torque\\_converter](http://en.wikipedia.org/wiki/Torque_converter)>

Wikipedia--Transmission (mechanics). [online]. [Accessed 23 Sep 2010]. Available from World Wide

Web: <[http://en.wikipedia.org/wiki/Transmission\\_\(mechanics\)](http://en.wikipedia.org/wiki/Transmission_(mechanics))>

WILLIS, Robert. 1870. *Principles of Mechanism, Second Edition*. London: Longmans, Green and Co.

ZACKRISSON, Tomas. 2003. *Modeling and simulation of a driveline with an automatic gearbox*.

ZF FRIEDRICHSHAFEN AG. 2006. *Technical Manual ZF-ECOMAT 4*. Germany: ZF Friedrichshafen AG , MC-C.

## 11 Appendix

### 11.1 Simscape MATLAB Supported Functions

Name	Restrictions	Discontinuous
plus		
uplus		
minus		
uminus		
mtimes		
times		
mpower		
power		
mldivide	Nonmatrix denominator	
mrdivide	Nonmatrix denominator	
ldivide		
rdivide		
eq	Do not use with continuous variables	
ne	Do not use with continuous variables	
lt		
gt		
le		
ge		
and		Yes
or		Yes
sin		
cos		
tan		
asin		
acos		
atan		
atan2		
log		
log10		
sinh		
cosh		
tanh		
exp		
sqrt	For negative numbers, calculated as $\sqrt{-1}$ . For example, $\sqrt{-1} = -1$ .	
abs		Yes
logical		Yes
sign		Yes

## 11.2 Complete Simscape code

### 11.2.1 The Ideal Gear model

```
component Ideal_gear
% Ideal Gear Description

nodes
    I = foundation.mechanical.rotational.rotational; % I:left
    O = foundation.mechanical.rotational.rotational; % O:right
end

parameters
    ratio = { 1, '1' }; % Min Gear ratio
end

variables
    t_in = { 0, 'N*m' };
    t_out = { 0, 'N*m' };
    w = { 0, 'rad/s' };
end

function setup
    through( t_in, I.t, [] );
    through( t_out, O.t, [] );
    across( w, I.w, O.w );

    % Parameter range checking
    if ratio <= 0
        pm_error( 'Ratio value must be greater than zero' );
    end
end

equations
    I.w == ratio * O.w;
    t_out == -ratio * t_in;
end

end
```

### 11.2.2 The Clutch model

```
component custom_clutch_test1
% custom_clutch_test1
nodes
    B = foundation.mechanical.rotational.rotational; % B:left
    F = foundation.mechanical.rotational.rotational; % F:right
end

inputs
    P = { 0, 'Pa' }; % :left
end

variables
    t = { 0, 'N*m' };
    w = { 0, 'rad/s' };
end

parameters
    R = { 0.1, 'm' }; % Radius
    N = { 6, '1' }; % Number of friction surfaces
    Uk = { 0.36, '1' }; % Kinetic friction constant
    W_tol = { 0.03, 'rad/s' }; % Tolerance level for clutch locking
end

parameters(Access=private)
    A = { 0, 'm^2' };
end
```

```

function setup

across( w, B.w, F.w );
through( t, B.t, F.t);

A=pi*R^2*N;
end

equations

if ((w>=W_tol) || (w<=-W_tol))    %Dynamic friction (SLIDING)
    t==2/3*R*P*Uk*A*sign(w);

else

    t==2/3*R*P*Uk*A/W_tol*w;

end

end
end

```

### 11.2.3 The Torque Converter model

```

component Torque_C
% Torque_C
nodes
    B = foundation.mechanical.rotational.rotational; % B:left
    F = foundation.mechanical.rotational.rotational; % F:right
end

inputs
    K = { 0, '1' }; % K:left
    Tr = { 0, '1' }; % Tr:left
end
outputs
    Rw= { 0, '1' }; % Rw:right
end

variables
    w = { 0, 'rad/s' };
    i = { 0, 'N*m' };
    t = { 0, 'N*m' };
end

parameters

    W_min = { 0.03, 'rad/s' }; % Tolerance level for torque converter
    D = { 0.3, 'm' }; % Circum diameter in m
    p = { 870, 'kg/(m*m*m)' }; % Density of medium (870 kg/m^3 for hudraulic fluid)
end

function setup

across( w, B.w, F.w );
through( i, B.t, [] );
through( t, F.t, [] );

end

equations

Rw==(F.w/B.w);

if ( ((B.w>=W_min) || (B.w<=-W_min) || (F.w>=W_min) || (F.w<=-W_min)) )

```

```

i==K*D^5*B.w^2*p*(1-(F.w/B.w));
t==i*Tr;

else
i==K*D^5*W_min^2*p*(1-(F.w/B.w));
t==i*Tr;

end

end

end
end

```

### 11.2.4 The Planetary Gear model

```

component plant_gear1
% plant_gear1
nodes
    C = foundation.mechanical.rotational.rotational; % C:right
    R = foundation.mechanical.rotational.rotational; % R:left
    S = foundation.mechanical.rotational.rotational; % S:left

end

parameters
    sgratio = { 5, '1' }; % Gear ratio
end

variables
    S_in = { 0, 'N*m' };
    C_in = { 0, 'N*m' };
    R_in = { 0, 'N*m' };
end

function setup
    through( S_in, S.t, [] );
    through( C_in, C.t, [] );
    through( R_in, R.t, [] );

    % Parameter range checking
    if sgratio == 0
        pm_error('simscape:NotZero','Gear ratio')
    end
end

equations

    %Overall kinematics
    (1+sgratio)*C.w==S.w+sgratio*R.w;

    %Torque balance
    R_in==sgratio*S_in;
    0==C_in+S_in+R_in;

end
end

```

## 11.3 Parameters setting used

### 11.3.1 Clutches

All clutches in the model have the same parameter settings and all clutch are actuated with a input pressure signal

Parameter	Value
Radius	0.3
Number of friction surfaces	6
Kinetic friction constant	0.4
Tolerance level for clutch locking	1e-3
Clutch pressure signal	45000

### 11.3.2 Torque Converter

Parameter	Value
Circum diameter	0.1
Density of medium	870
Tolerance level for torque converter	0.03
K-factor	[45.3 45.2 45.2 45.4 45.5 46.2 47.7 50.8 56.1 60.7 70.0 93.7]
Torque ratio	[1.982 1.894 1.804 1.692 1.571 1.440 1.303 1.159 1.000 0.995 0.995 0.995]

### 11.3.3 Retarder

Parameter	Value
Ratio	1.6
Percent	0.8
Engine friction	0.1
Retarder ratio	[2200 2200 2200 1700 1300 1000]

### 11.3.4 Planetary gears

#### 11.3.4.1 Planetary Gear 1

Parameter	Value
Ring/Sun ratio	2.405

#### 11.3.4.2 Planetary Gear 2

Parameter	Value
Ring/Sun ratio	2.4396

#### 11.3.4.3 Planetary Gear 3

Parameter	Value
Ring/Sun ratio	2.43

### 11.3.5 Final gear

Parameter	Value
Ratio	6.2

### 11.3.6 Automatic Logic Program

Parameter	Value
Up-shifting	150
Down-shifting	110
After	0.5

### 11.3.7 Air drag

Parameter	Value
Air drag coefficient	0.65
Cross section area	10
Air density	1.225

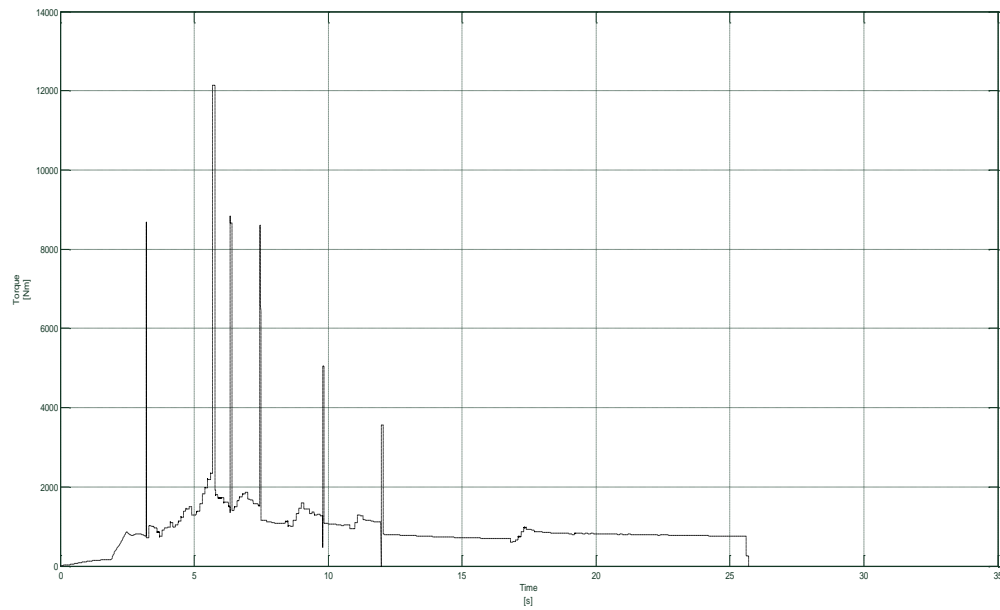
### 11.3.8 Rolling resistance

Parameter	Value
Rolling resistance coefficient 1	50
Rolling resistance coefficient 1	0.2
Mass	11440

### 11.3.9 Other parameters

Parameter	Value
Tire Radius	0.507
Vehicle Effective Inertia	2462
Engine Inertia	4
Impeller Inertia	1.74
Turbine Inertia	1.12
Gearbox Inertias	1.5e-3
Propeller shaft Inertia	1

## 11.4 Torque output on propeller shaft



The figure above illustrates the torque output on propeller shaft. The first peak represents the torque transferred during torque converter when the lock-up clutch is activated. The remaining peaks represent the torque transferred during each gear shift. Each peak lasts approximately between 0.05 – 0.1s.

## 11.5 The final validation concept

